NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

DENOISING OF OCEAN ACOUSTIC SIGNALS USING WAVELET-BASED TECHNIQUES

by

Robert J. Barsanti, Jr.

December, 1996

Thesis Advisor: Monique P. Fargues
Co-Advisor: Ralph Hippenstiel

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1996	Master's Thesis

4. TITLE AND SUBTITLE : DENOISING OF OCEAN ACOUSTIC SIGNALS USING WAVELET BASED-TECHNIQUES	5. FUNDING NUMBERS
6. AUTHOR(S) Robert J. Barsanti, Jr.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis investigates the use of wavelets, wavelet packets and cosine packet signal decompositions for the removal of noise from underwater acoustic signals. Several wavelet based denoising techniques are presented and their performances compared. Results from the comparisons are used to develop a wavelet-based denoising algorithm suitable for a wide variety of underwater acoustic transients. Performances of the denoising algorithm are compared to those of a short-time Wiener filter implementation, and demonstrate that wavelet-based methods are a viable tool for the denoising of acoustic data.

14. SUBJECT TERMS Wavelets, Denoising, Cosine Packet Transform, Acoustic Signals.			15. NUMBER OF PAGES 115 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

DENOISING OF OCEAN ACOUSTIC SIGNALS USING WAVELET-BASED TECHNIQUES

Robert J. Barsanti, Jr.
Lieutenant Commander, United States Navy
B.S., Polytechnic Institute of New York, 1982

Submitted in partial fulfillment
of the requirements for the degrees of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
MASTER OF SCIENCE IN ENGINEERING ACOUSTICS

from the

NAVAL POSTGRADUATE SCHOOL
December 1996

Author: Robert J. Barsanti, Jr.

Approved by: Monigue P. Fargues, Thesis Advisor

Ralph Hippenstiel, Co-Advisor

Robert M. Keolian, Chairman, Engineering Acoustics Academic Committee

Herschel H. Loomis, Jr., Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This thesis investigates the use of wavelets, wavelet packets and cosine packet signal decompositions for the removal of noise from underwater acoustic signals. Several wavelet based denoising techniques are presented and their performances compared. Results from the comparisons are used to develop a wavelet-based denoising algorithm suitable for a wide variety of underwater acoustic transients. Performances of the denoising algorithm are compared to those of a short-time Wiener filter implementation, and demonstrate that wavelet-based methods are a viable tool for the denoising of acoustic data.

TABLE OF CONTENTS

I. INTRODUCTION

Wavelet analysis provides a unified framework to a number of techniques that are applied in various research areas including mathematics, computer imaging and geophysics. In signal processing wavelet-based techniques can be found in applications such as multi-resolution processing, signal compression, subband coding and noise removal.[1]

Wavelets and their close relatives, wavelet packets and cosine packets provide a valuable tool in the removal of noise primarily because they provide a large variety of flexible basis functions that allow for projection of the signal into a coordinate system, in which the signal characteristic are distinguishable from that of the noise. Wavelet-based transforms also permit analysis of data at multiple levels of resolution. This multi-resolution property makes these transforms unique, and permits the user to zoom-in on local signal features or zoom-out to take global views of the signal to be analyzed. This ability is particularly relevant to the types of acoustic signals studied in this thesis, which are composed of both short duration broadband transients that are highly localized in time and tonals or narrowband signals that are localized in frequency.

The wavelet thresholding methods used in this study are an extension to, and combination of, methods found throughout the signal processing literature. In particular we apply denoising techniques originally proposed by Donoho et. al. [2], to underwater acoustic transients in low signal to noise ratio (SNR), and colored noise environments.

The thesis has seven additional chapters. Chapter II is an introduction to ocean noise, its sources and characteristics. Chapter III discusses the theory of wavelet analysis

and its extension to the wavelet packet transform. Chapter IV provides a brief treatment of the local trigonometric and the cosine packet transforms. Chapter V describes the criterion used for basis selection and the Best Basis algorithm. In Chapter VI the principles and application of wavelet denoising are discussed. Various thresholding methods are examined and contrasted here as well. Chapter VII discusses the results of applying wavelet-based methods to ocean acoustic transients. Chapter VIII provides a summary and conclusions.

II. NOISE

Noise can be considered the undesired part of the input, whereas the signal is the desired part of the input. The presence of noise at the input masks the signal and can hide its relevant features. Removal of the noise from the input aids in the detection, estimation, and classification of signals. In underwater acoustics, the noise can be separated into two broad categories. The first is background noise or ambient noise which is inherent to the ocean, and includes all natural and man-made acoustic sources that contribute to the underwater noise level in the absence of the receiver. The second category is self noise which arises from the receiving system and its platform.

The literature covering the field of underwater acoustic noise is massive and is the second most prolific topic in all of underwater acoustics (after sound propagation) [3]. This chapter briefly discusses the types of noise typically found in underwater acoustic data, its sources and its characteristics.

A. NOISE SOURCES

1. Ambient Noise

Ambient noise is the background noise found at a particular location of the ocean, and it is independent of the means used to measure it. The wind and its effect on the ocean surface contribute to this noise across the entire frequency spectrum. The ambient noise spectrum is shown in Figure 2.1. It displays the sound pressure level (SPL) found in a one Hz frequency band in decibels (dB) at 1 meter referenced to 1 μPa.

As the shape of the curves suggest, different processes are responsible for the generation of the noise in different frequency ranges. In the frequency range below about

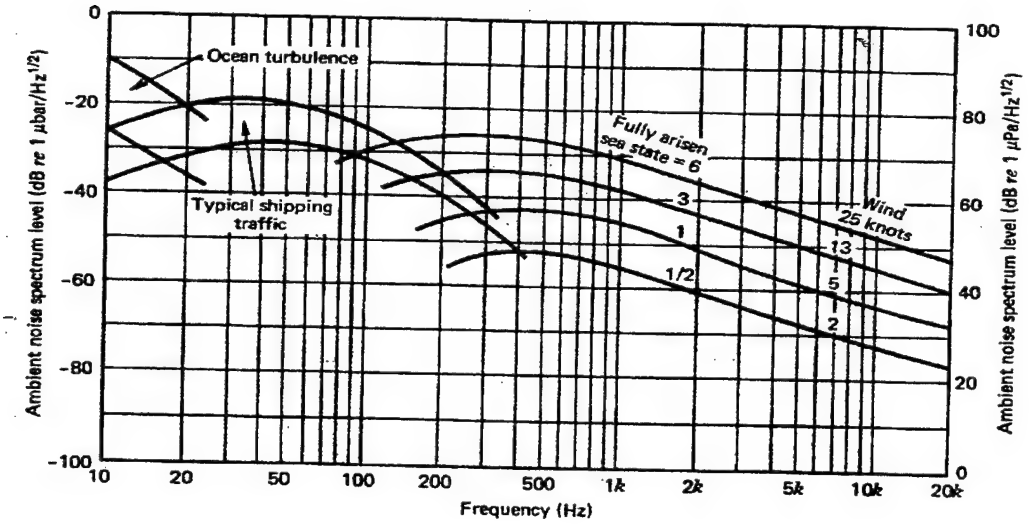

Figure 1.1: Ambient ocean noise spectrum. From Ref. [4].

10 Hertz (Hz), ambient noise is primarily caused by ocean turbulence and seismic activity. This turbulence is attributed to the wind and wave action at the ocean surface. The noise spectrum falls off rapidly in this region with a slope of about 10 dB per octave. In the frequency range between 10 and 200 Hz, distant shipping noise dominates the spectrum. In areas of low or no shipping noise, the ever present wind and the resulting surface action still generate acoustic noise in this range. Other man made sources such as seismic exploration and oil production are also found in this band and contribute to a lesser degree.

Above 100 Hz, shipping noise begins to fall off rapidly, and the agitation of the local sea surface due to wind turbulence, surface motion, wave interaction, spray, and cavitation becomes important. These sea surface effects begin to prevail somewhere between 200 and 500 Hz, depending on the local shipping density and the current wind

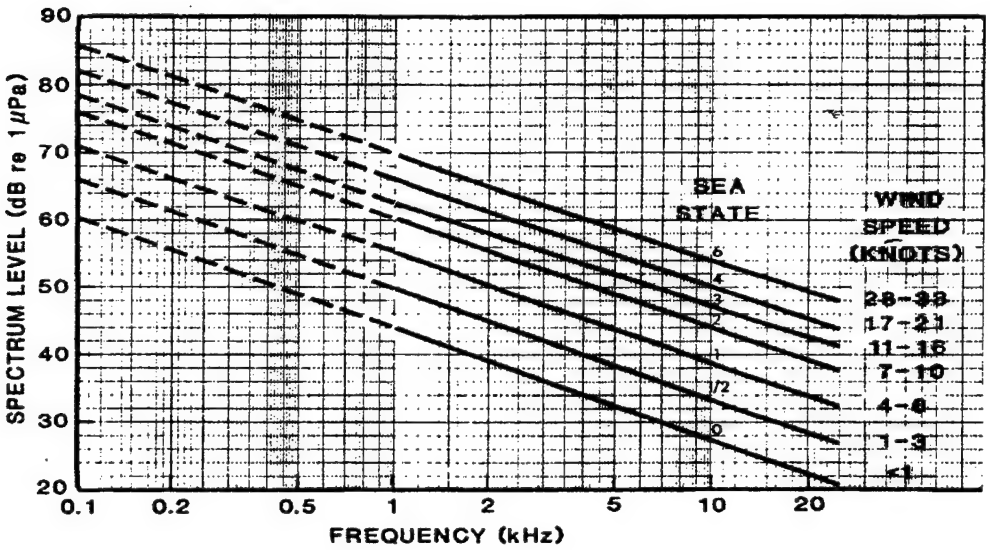

Figure 2.2: Knudsens curves. From Ref. [3].

speed and sea state. The first published study of ocean noise in the band from 200 - 2000 Hz was made by V.O. Knudsen and produced the curves shown in Figure 2.2. These curves remain accurate today for frequencies greater than 1000 Hz. (Between 200-800 Hz, the spectrum is nearly flat contrary to Knudsens' predictions) [3]. As these curves show, above one KHz the noise decays at slightly less than four dB per octave, and this trend continues up to about 50 KHz.

In the frequency range above 50 KHz thermal noise begins to dominate the background noise. This results in an increase in the noise level at about six dB per octave, and it attributed to the thermal agitation of the water molecules.

2. Self Noise

Self noise is the noise associated with the receiving system and its platform. The most severe form of self noise typically results from the flow of water past the face of the hydrophone on a moving ship. This flow noise is speed dependent and can easily exceed

that of the ocean ambient noise at higher ship speeds. However at slow speeds, or for stationary arrays, self noise is usually less important than ambient noise [4]. Machinery noise and propeller noise are other sources of self noise. They both produce vibrations that can be sent to the recording system via the mechanical structure of the ship, or can generate acoustic waves received at the hydrophone.

On a stationary platform, ocean currents cause noise by carrying water of varying turbulence and temperature in contact with the hydrophone. This creates noise due to the piezoelectric and pyroelectric sensitivity of the hydrophones and is called pseudo-noise. Another source of noise is caused by cable strumming which can occur if the hydrophone is mounted from a flexible cable. [3]

Self noise is measurable and efforts can be (and usually are) made to minimize it. Ships can reduce their speed to limit flow induced noises and propeller noise. Efforts are made to isolate machinery sounds and to reduce machinery vibrations. Hydrophones are designed with thermal insulators to reduce pyroelectric effects. Numerous other design issues of the recording system and its platform are considered to limit the effect of self noise. Even in the face of all these measures taken, one would be negligent to ignore the possible presence of self induced noises in any data analyzed.

B. PROPAGATION EFFECTS

The transmission of acoustic energy in the ocean is largely affected by the properties of the water mass encountered by the acoustic wave. The distribution of water temperature, salinity, and density along with the depth of the water and bottom structure determine the transmission path of sound in the ocean. Acoustic energy can be reflected

and scattered from both the surface and bottom of the ocean, permitting the possibility of more than one transmission path for a single signal.

The attenuation of acoustic energy in the ocean medium is dependent on its frequency. The general result is that high frequencies are attenuated more than low frequencies. This causes the ocean to act like a low pass filter, and is responsible for the higher levels of lower frequency ambient noise.

The ocean is not isotropic, implying that the noise is not homogenous in all directions, nor at all depths at a given geographical location. The combination of these propagation effects adds to the variability of ocean acoustic noise, and makes it extremely difficult to predict ocean noise levels. As a result the characteristics of the noise at the receiving platform will not only depend on the actual sources but also on the properties of the ocean itself along the transmission path of the acoustic energy.

C. NOISE CHARACTERISTICS

The frequency spectrum of ambient noise shown in Figure 2.1 makes it clear that ocean noise is colored. This fact complicates the filtering (noise removal) problem since the exact color of the noise is uncertain. One solution to dealing with colored noise is to apply a pre-whitening transform to the data before attempting to remove the noise. Pre-whitening will be discussed further in Chapter VII.

Ambient ocean noise changes over time and is therefore non-stationary. However the variability of the predominant sources (wind speed and shipping density) change slowly over the course of hours or longer. Similarly the properties of the ocean itself that affect propagation (such as temperature and density) change even more slowly. So for the

purpose of analyzing data segments on the order of a few seconds, the ambient ocean noise can be assumed to be stationary [3].

For short periods of time on the order of fractions of a second to a few minutes, ambient ocean noise is also known to have a Gaussian amplitude distribution [3]. This statistical property of the noise is extremely important in devising a method to remove it from the input, because it lends it self to analysis using well known mathematical descriptions and models.

D. NOISE IN DATA

The purpose of this section is to display the frequency and amplitude distribution characteristics of the noise found in the data sets analyzed in this study. The goal is to apply simple tests to verify that the earlier assumptions of Gaussian colored noise are reasonable.

Two samples of the noise were extracted from two different data records, by extracting portions of the data prior to the known onset of the signal. In each case 4096 points of digitized data were extracted, and the maximum amplitude was normalized to unity.

Plots of the power spectrum magnitude for each of the noise samples is shown in Figure 2.3. The spectrum of each of these noise samples was plotted using the Matlab® power spectral density (*psd.m)* function [5]. The function parameters used were an FFT length of 256 points and a rectangular window, with no overlap of adjacent segments. From the figure we can see that neither sample is white and that the color of each is different.

A histogram of each noise sample was performed to display the amplitude distribution of the noise samples. The 4096 points were separated into 32 bins of 28 points each. These histograms are shown in Figure 2.4. Both samples appear to have a Gaussian distribution. Although more rigorous tests are possible, a further indicator of the Gaussian nature of the noise can be displayed from a *normal probability plot* of the data. This plot compares the percentiles of the sample data to the corresponding percentiles of a normal distribution. If the sample data is normally distributed the points in the plots should lie along a straight line [6]. From the normal probabilty plots shown in Figure 2.5, we can observe that the noise sample data is essentially Gaussian distributed. These simple tests allow us to reasonably conclude that the analyzed samples consist of Gaussian distributed colored noise.

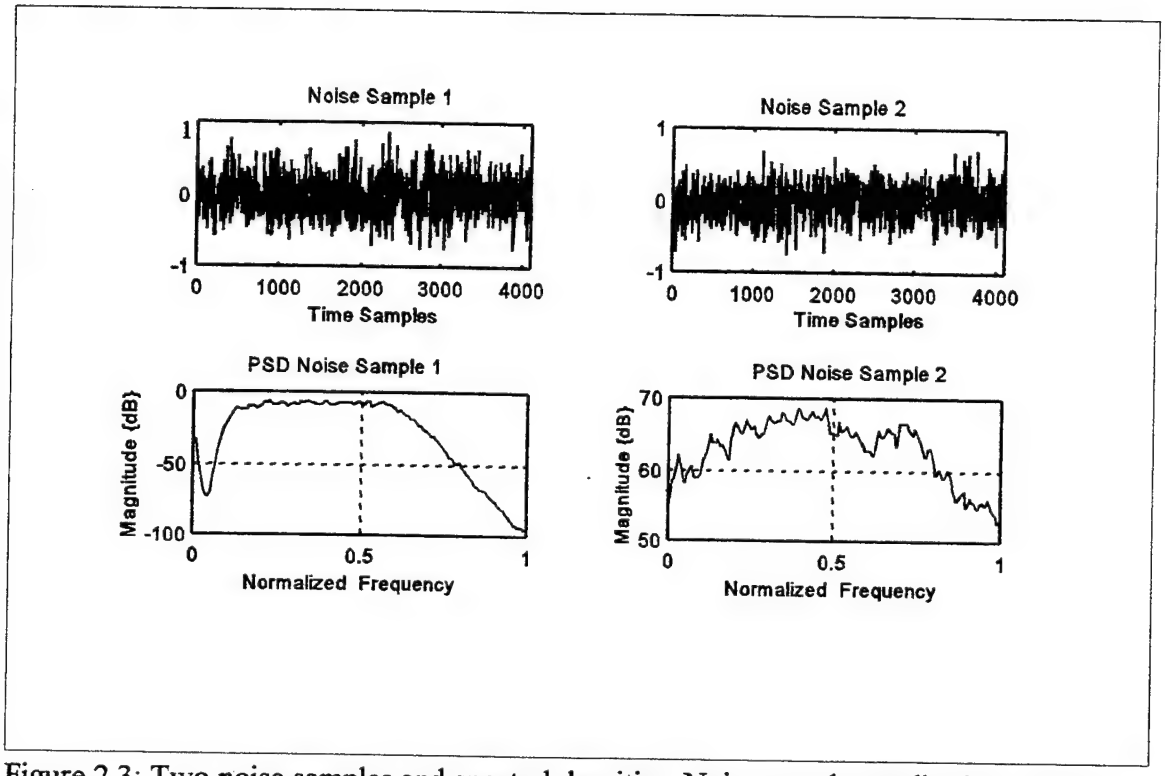

Figure 2.3: Two noise samples and spectral densities. Noise sample amplitude vs. time samples (top). Noise power spectral densities vs. normalized frequency (bottom).

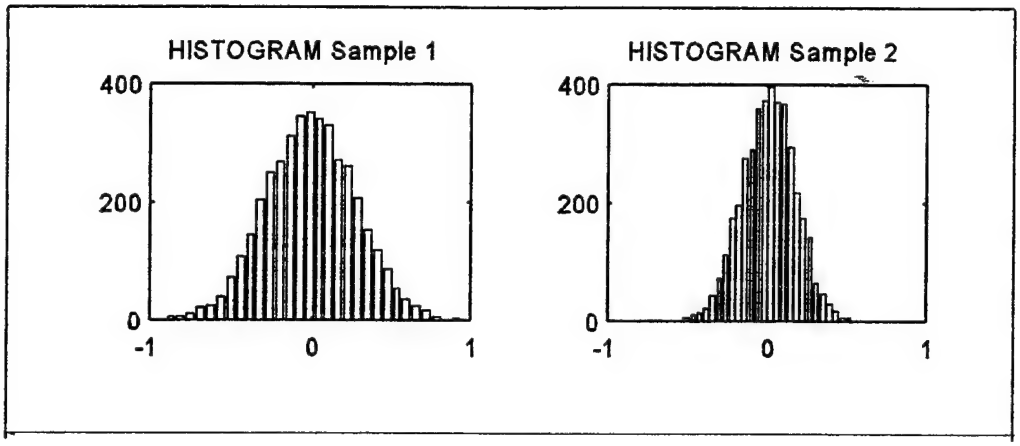

Figure 2.4: Noise Histograms.

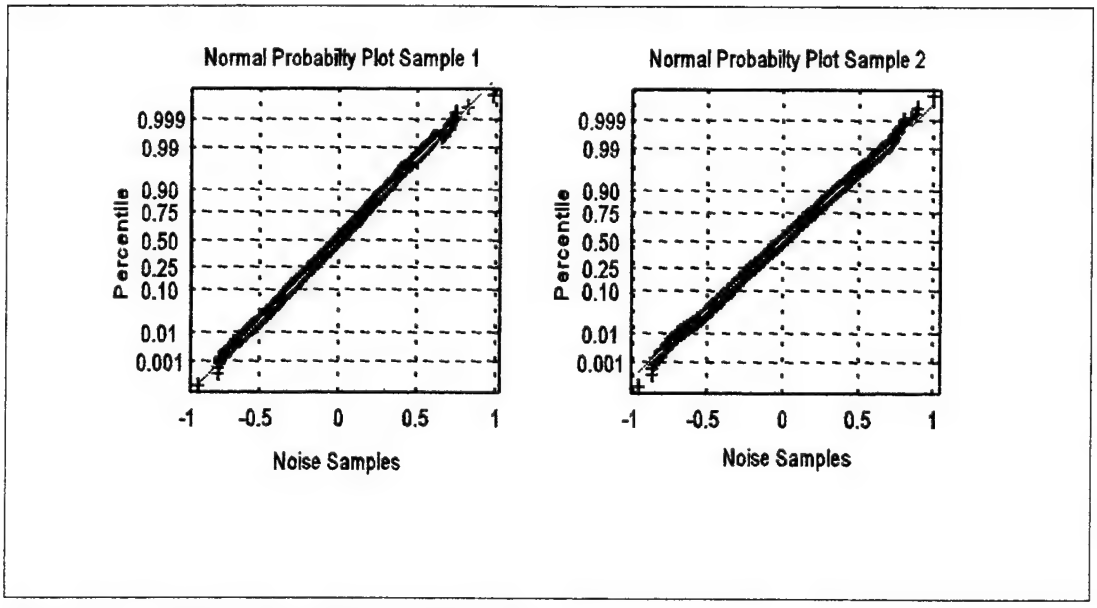

Figure 2.5: Noise Normal Probability Plots.

III. WAVELETS AND WAVELET PACKETS

Wavelet analysis is easiest to understand as an extension of the more familiar Fourier analysis. Therefore this chapter first discusses Fourier analysis before introducing wavelet analysis.

A. FOURIER ANALYSIS

1. Fourier Series

Any periodic function $x_p(t)$ can be represented as the infinite summation of sine and cosine terms [7]. If the function $x_p(t)$ has period T_o, its Fourier series expansion can be written using a trigonometric form as:

$$x_p(t) = a_o + \sum_{n=1}^{\infty} [a_n \cos(2\pi n f_o t) + b_n \sin(2\pi n f_o t)] , \qquad (3.1)$$

where $f_o = 1/T_o$ is the fundamental frequency. The quantity nf_o represents the n^{th} harmonic of the fundamental frequency. The coefficients a_n and b_n represent the amplitudes of the cosine and sine terms at the n^{th} harmonic of the fundamental frequency. The coefficient a_o represents the mean value of the periodic signal $x_p(t)$ over one complete period. [7]

The Fourier series of Equation 3.1 can be equivalently written in terms of complex exponentials. Substituting the exponential forms of the sine and cosine terms into Equation 3.1 produces the equivalent complex exponential Fourier series given by:

$$x_p(t) = \sum_{n=-\infty}^{\infty} C_n \, e^{\,j2\pi n f_o t} \,. \qquad (3.2)$$

The parameters C_n are the complex Fourier coefficients, and represent the complex weights of the n^{th} harmonic of the complex basis function $\exp(j2\pi n f_o t)$ [7]. These coefficients can be found from the analysis equation given by:

$$C_n = \frac{1}{T_o} \int_{-\frac{T_o}{2}}^{\frac{T_o}{2}} x_p(t) \, e^{-j2\pi n f_o t} dt \qquad n = 0,\pm 1,\pm 2.... \qquad (3.3)$$

The complex exponential analysis equation provides the coefficients necessary to exactly reconstruct the periodic signal from its Fourier series expansion. A plot of the magnitude of C_n versus frequency is called the magnitude spectrum of the signal $x_p(t)$. The spectrum provides a frequency domain presentation of the signal.

2. Fourier Transform

The Fourier transform of a general continuous function $x(t)$ is defined as:

$$X(f) = \int_{-\infty}^{\infty} x(t) \, e^{-j2\pi f t} \, dt \,. \qquad (3.4)$$

$X(f)$ is a continuous function of the frequency variable f. Equation 3.4 is directly analogous to Equation 3.3, and in fact can be derived from it by representing $x(t)$ as a periodic function with infinite period and taking the limit as T_o goes to infinity [8]. The

magnitude spectrum of $x(t)$ is a plot of the magnitude of $X(f)$ versus frequency and is continuous. The original signal $x(t)$ can be recovered exactly from $X(f)$ by means of the inverse Fourier transform defined as:

$$x(t) = \int_{-\infty}^{\infty} X(f)\ e^{j2\pi ft}\ df\ . \qquad (3.5)$$

The two functions $x(t)$ and $X(f)$ uniquely define each other and are known as a Fourier transform pair.

3. Short-Time Fourier Transform

The Fourier analysis techniques described above provide a frequency domain presentation of the signal. These methods can be applied to signals whose frequency structure does not vary with time (i.e., stationary signals). When the signal is non-stationary, it is desirable to have a description that involves both time and frequency. [7]

The short time Fourier transform (STFT) can be viewed as an extension of the Fourier transform devised to map the signal into the two dimensional time-frequency plane. The STFT uses a sliding window function $g(t)$ to segment the signal into small uniform blocks of time. Each block is made short enough so that the signal may be considered essentially stationary within that segment. The Fourier transform is then applied to each time segment to produce the STFT representation given by:

$$S(\tau,f)=\int_{-\infty}^{\infty} x(t)\ g^{*}(t-\tau)\ e^{-j2\pi ft}dt\ , \qquad (3.6)$$

where $S(\tau,f)$ displays the evolution of the signals frequency information over time. A plot of the squared magnitude of $S(\tau,f)$ is called a spectrogram, and it provides a measure of the signal energy in the time - frequency plane.

Many different window functions $g(t)$ may be selected in practice, and the choice will effect the resulting STFT. Once a window function has been chosen its shape will determine the resolution of the time information ($\Delta\tau$) in the time - frequency plane. As a result of the uncertainty principle, the time resolution, and the frequency resolution (Δf) of a given signal are inversely related and their product has a lower bound of $1/4\pi$ [7]. This produces a trade-off of time resolution for frequency resolution. Since the choice of window will fix $\Delta\tau$ (and also thus Δf) over the entire signal length, the STFT partitions the time - frequency plane into a uniform grid. The drawback of this property is that both $\Delta\tau$ and Δf are fixed throughout the analysis of the signal, and can not simultaneously provide good time resolution (requiring short windows) and good frequency resolution (requiring long windows).

B. WAVELET ANALYSIS

1. The Continuous Wavelet Transform

In Fourier analysis the signal is decomposed into a series of different frequency sinusoids. Mathematically the STFT can be viewed as an inner product of the signal with a two parameter basis function given by $g(t-\tau)\, exp(-j2\pi f\, t)$ [7]. In wavelet analysis the signal is also decomposed into a family of two parameter basis functions $\Psi_{a,\tau}(t)$, (where $\Psi_{a,\tau}(t) = \Psi[(t-\tau)/a]$), with specific properties. These basis functions are called wavelets.

One advantage of wavelet analysis is that it allows selection of a wide variety of basis functions, as opposed to being restricted to the sinusoids of Fourier analysis. Two important characteristics of wavelets are that; 1) the wavelet function $\Psi(t)$ be of finite duration, and 2) the wavelet function $\Psi(t)$ have zero average value (like that of Fourier sinusoids). The second characteristic requires that the basis functions oscillate above and below zero, and gives rise to the name wavelet or small wave [9]. Although there are numerous functions that meet the necessary properties to be classified a wavelet only a few classes have thus far been shown to be of general interest in signal processing. The Haar, Daubechies, Coiflet, and Symmlet are a few of the more popular classes and are shown in Figure 3.1.

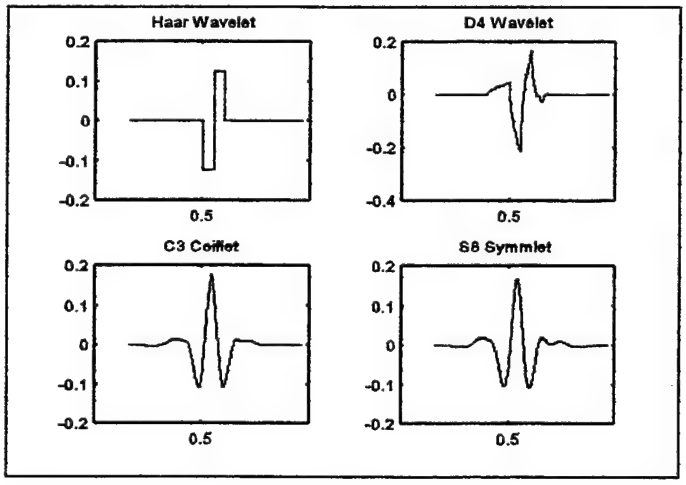

Figure 3.1: Four wavelets in the time domain. From Ref. [10].

The continuous wavelet transform (CWT) of a signal $x(t)$ is defined as:

$$C(\tau,a)=\frac{1}{\sqrt{a}}\int x(t)\Psi^{*}(\frac{t-\tau}{a})dt\ , \qquad (3.7)$$

where $\Psi(t)$ is the analysis wavelet. The parameter τ denotes translation in time, and the scale factor a denotes dilation in time. The factor $1/\sqrt{a}$ normalizes the energy of the CWT. The scale factor in wavelet analysis plays an analogous role to inverse frequency in Fourier analysis. For example, consider the signal $x(t) = cos\ (t/a)$, where a denotes the scale factor. If a is made larger the function $x(t)$ will oscillate slower in time, and is thus expanded or stretched. If a is made smaller $x(t)$ will oscillate faster, and is therefore contracted in time. The scale factor a is therefore a method to expand or contract the analysis wavelet in the time domain [9]. (Recall that this will have the opposite effect on the analysis wavelet in the frequency domain).

The time resolution and frequency resolution of the CWT is also controlled by the scale factor. Low scales (small values of a) correspond to high frequency wavelets and provide good time resolution. High scales (large values of a) correspond to low frequency wavelets with poor time resolution but good frequency resolution. Figure 3.2 displays the Symmlet 8 wavelet for decreasing values of the scale factor a, in both the time and frequency domain. It is clear from the figure that, as a decreases, thinner (more localized) time domain wavelets and fatter (less localized) frequency domain wavelets are produced.

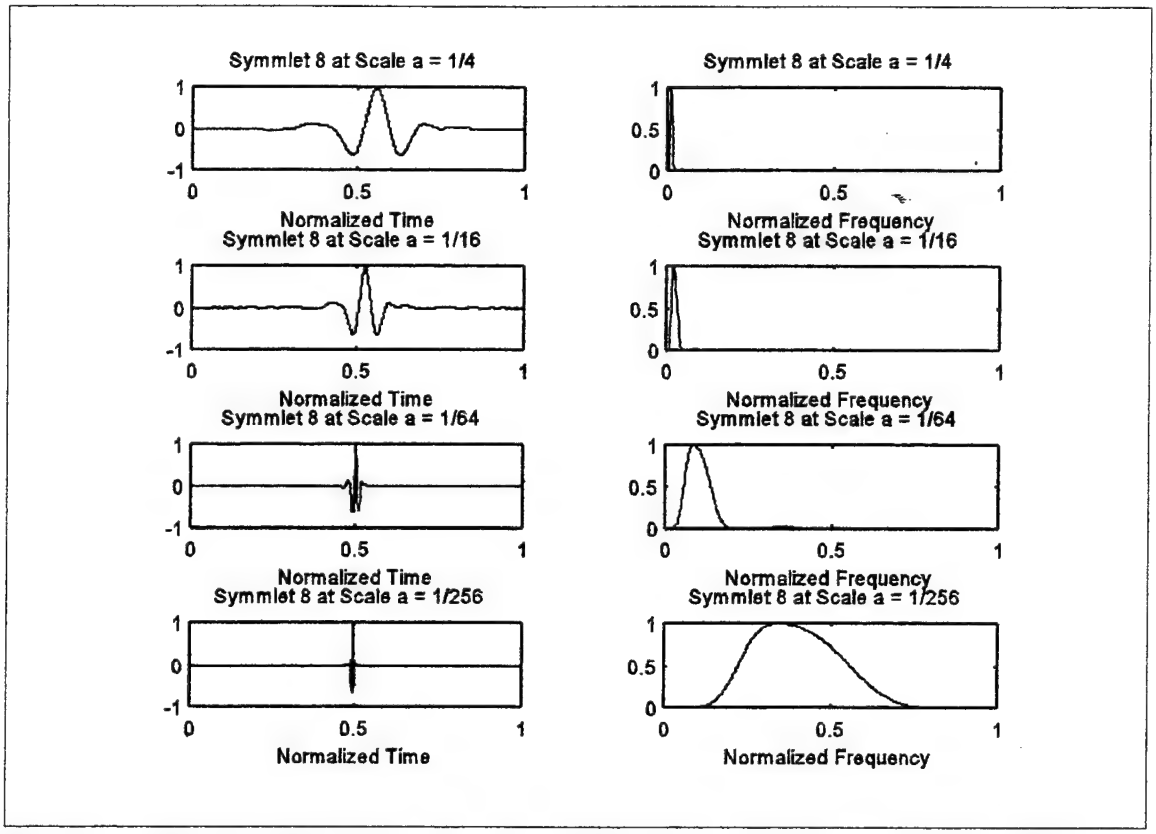

Figure 3.2: Symmlet 8 wavelet in the time and frequency domains as a function of the scale parameter a. The scale factor a decreases from the top to bottom plots. After Ref [10].

So in summary, large values of a mean high scales, low frequencies, good frequency resolution, and poor time resolution.

A second advantage of wavelet analysis is the multiresolution capability it provides in the time - frequency plane. A comparison of the time - frequency mapping of the STFT and the CWT is shown in Figure 3.3. The STFT produces a uniform grid with a constant time resolution and frequency resolution, while the CWT has time resolution and frequency resolution that depend on the scale. Note that the CWT time resolution improves at higher frequency and the frequency resolution degrades.

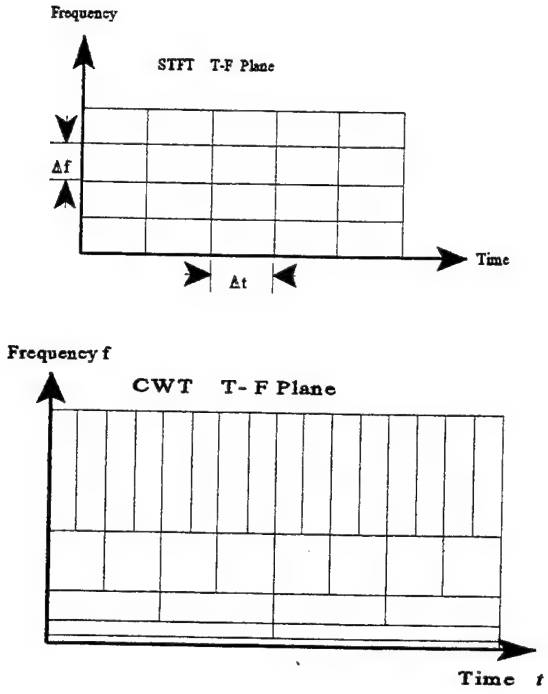

Figure 3.3: Time - Frequency plane for STFT and CWT.

Similar to the STFT spectrogram, the CWT *scalogram* is defined as the squared magnitude of $C(\tau,a)$, and it is a measure of the energy of the signal in the time - scale plane [1]. Further insight to the multiresolution capability of the CWT can be gained by comparing the influence of signals in the time - scale plane. Figure 3.4 shows a comparison of the regions of influence of the spectrogram and scalogram for two different signals. The top plots display an impulse function at $t = t_o$. Note that the scalogram permits a narrow time localization of this signal in the low scale portion of the plot. The lower plots display the regions of influence for a signal composed of two sines at frequencies f_1 and f_2. Note the CWT has better frequency resolution at high scales and poorer frequency resolution at low scales.

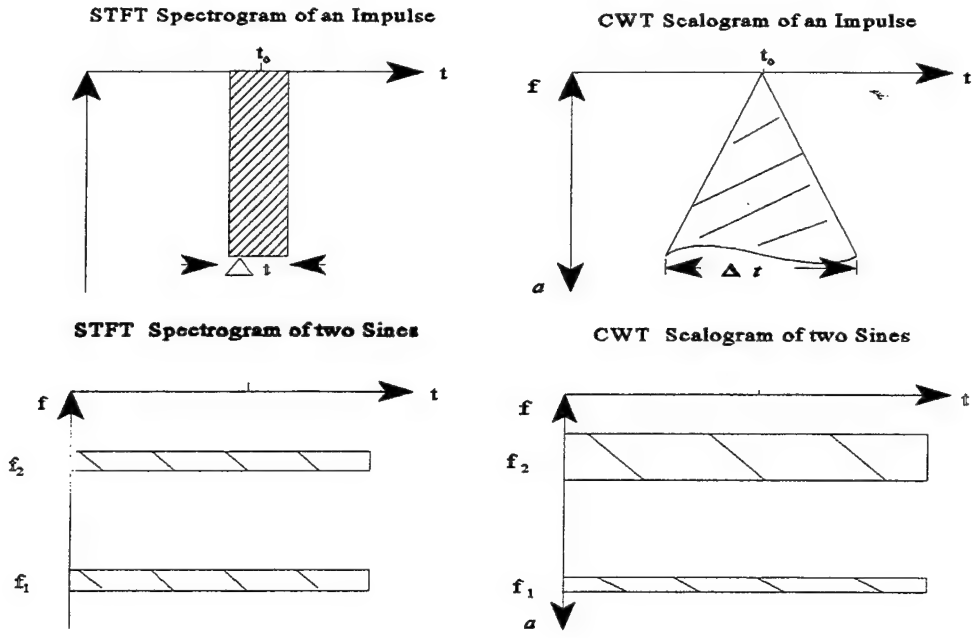

Figure 3.4: Spectrograms and Scalograms for two signals. Top plots display transforms for an impulse function. Bottom plots display transforms for two sines. After Ref [1].

2. The Discrete Wavelet Transform

The discrete wavelet transform (DWT) is defined by restricting the scale and time parameters of the CWT to discrete values. The DWT of a discrete signal $x(n)$ is defined by:

$$C(a,b) = \sum_{n=1}^{N} \frac{1}{\sqrt{a}} x(n)\ \Psi^*\left(\frac{n-b}{a}\right), \qquad (3.8)$$

where a, b, n are the discrete versions of a, τ, and t of Equation 3.7 respectively. The

scaling factor is further restricted to be given by:

$$a = a_o^J \qquad J = 0,1..... \qquad (3.9)$$

The choice of a_0 will govern the accuracy of the signal reconstruction via the inverse transform. It is popular to choose $a_0 = 2$, because it provides small reconstruction errors and permits for the implementation of fast algorithms [1]. Setting $a = 2^J$ produces octave bands called dyadic scales. At each scale as J increases, the analysis wavelet is stretched in the time domain, and compressed in the frequency domain by a factor of two. (See Figure 3.2). The result being that the DWT output at each dyadic scale J produces more precise frequency resolution and less precise time resolution.

Also note that as J increases the translation term $b/2^J$ becomes smaller, and thus b must necessarily increase to cover all translations. The result is that the DWT output grows in length by a factor of two at every scale. This produces extremely large DWT vectors at the higher scales. This computational difficulty can be alleviated by realizing that at each successive octave, the DWT output contains information at half the bandwidth compared to that of the previous scale, and thus can be sampled at half the rate according to Nyquists' rule [10]. This decimation (or subsampling) is accomplished mathematically by restricting values of the shift parameter b. Letting $b = k \cdot 2^J$ where k is an integer, and replacing a by 2^J yields the decimated DWT given by:

$$C(2^J,k2^J) = \sum_{n=1}^{N} \frac{1}{\sqrt{a}} x(n)\ \Psi^*(2^{-J}n-k)\ , \qquad (3.10)$$

where $J = 0,.....\log_2(N)$ and $k = 1,.....N \cdot 2^{-J}$. The term $k \cdot 2^J$ in the argument of the DWT, indicates that $C(a,b)$ is decimated by a factor of two at each successive scale J by retaining only the even points. The resulting DWT coefficients form a [J x k] matrix where each element represents the similarity between the signal and the analysis wavelet at scale J and shift k. It is common practice therefore to rewrite Equation 3.10 explicitly in terms of the parameters J and k, leading us to the decimated DWT equation defined as:

$$C_{j,k} = \sum_n \sqrt{1/2}^J \, x(n) \, \Psi^*(2^{-J}n - k) \; . \qquad (3.11)$$

The Symmlet 8 wavelet is shown at various scales J and shifts k in Figure 3.5.

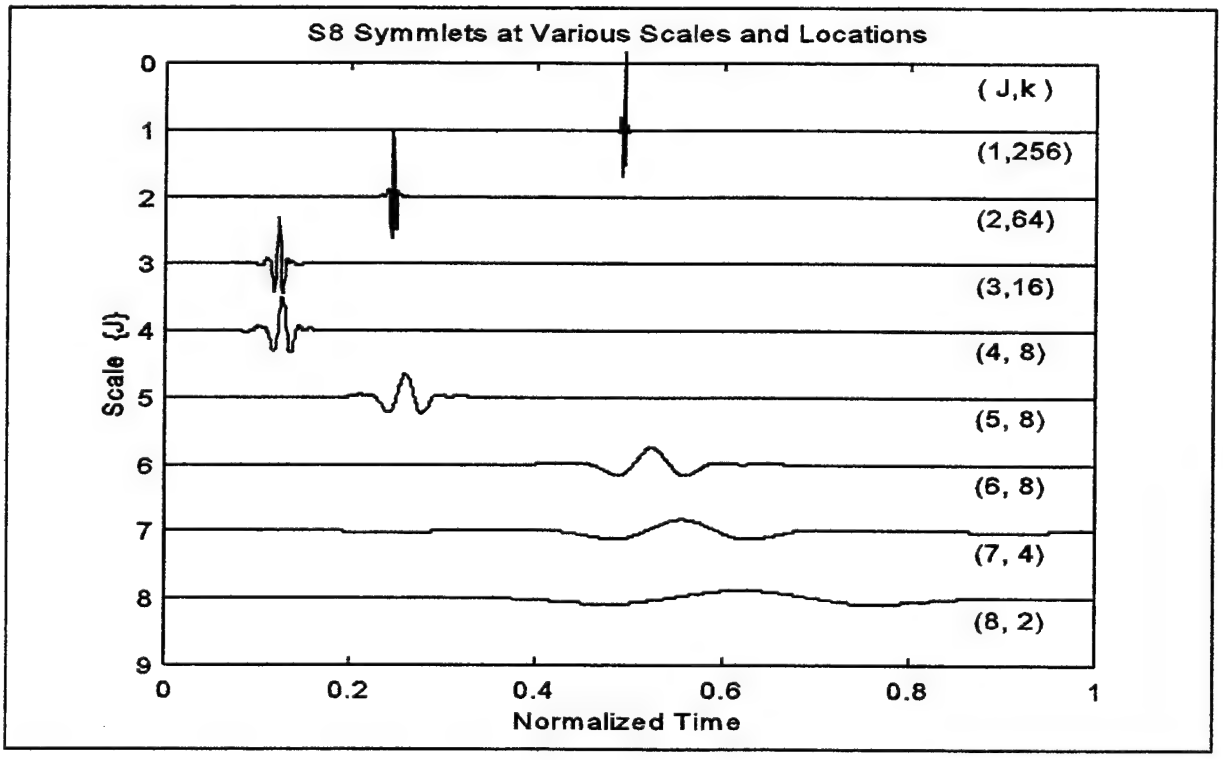

Figure 3.5: Symmlet 8 wavelet at various scales J and positions k.

3. Implementing the Discrete Wavelet Transform

An efficient way to implement the DWT of Equation 3.11 using filters was developed by Mallat [11]. This scheme uses a complementary pair of lowpass (LP) and highpass (HP) filters. These filters equally partition the frequency axis and are known as quadrature mirror filters (QMF) [12]. Figure 3.6 displays the filter arrangement and the frequency response characteristics of the QMF. The output of the HP filter contains the details of the signal while the output of the LP filter contains the rough shape of the signal. Since each filter output covers only half the original frequency range of the input, each can be decimated by a factor of two by retaining only the even points. The combined decimated output of the two filters is a data set which comprise the DWT coefficients at the first scale. This process is repeated on the LP filter output to produce further decomposition of the signal into LPHP and LPLP parts at the next scale. The filtering and decimating operations can be continued until the number of samples is reduced to two. At each successive iteration (scale) the frequency range of the output is reduced in half by the LP filter, and the frequency resolution is improved by the decimation. Figure 3.7 shows how a data set of 2^J samples can be decomposed to produce a maximum of J levels of transform coefficients. Figure 3.8, displays the resulting transformed coefficients in a tree structure. Note that movement down the tree relates to lower frequency (higher scale) coefficients.

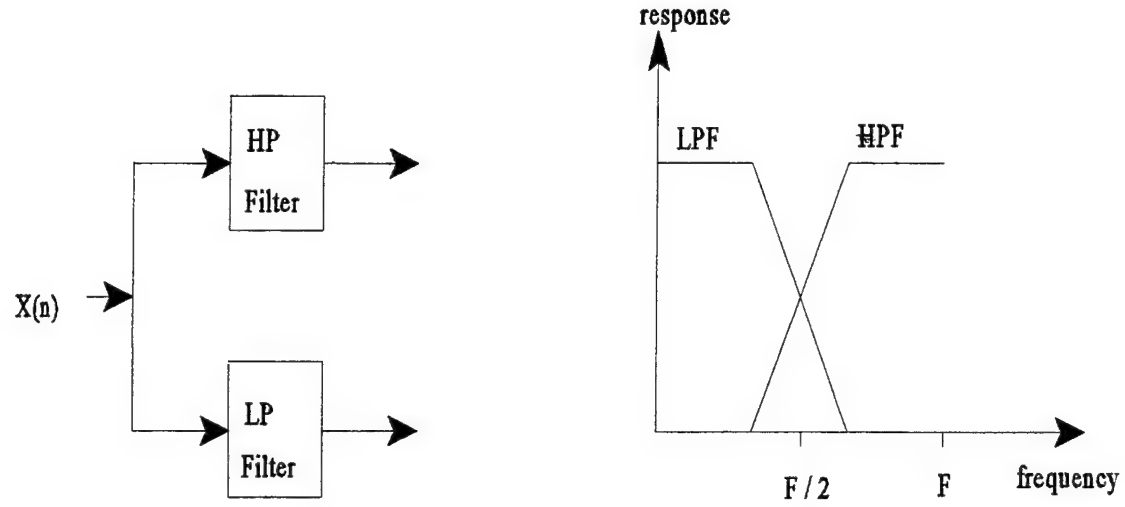

Figure 3.6: Schematic representation of quadrature mirror filters.

The decimated DWT described above will produce an orthogonal decomposition of the input signal only if the QMF pairs (i.e., the wavelets) are properly chosen. Such filter pairs have to possess specific mathematical properties and exhibit restrictive symmetry characteristics.[13]

Although the DWT filtering operations are linear and time invariant, the decimation combined with the filtering results in a time-variant system. Recall, that a time variant system implies that shifts in the system input will not produce an equivalent shift in the system output [12]. In fact, a shift of even a few samples in the signal's starting point can completely change the wavelet decomposition coefficients. This difficulty complicates the performance of signal detection, feature extraction, and classification in the wavelet transform domain [14,15].

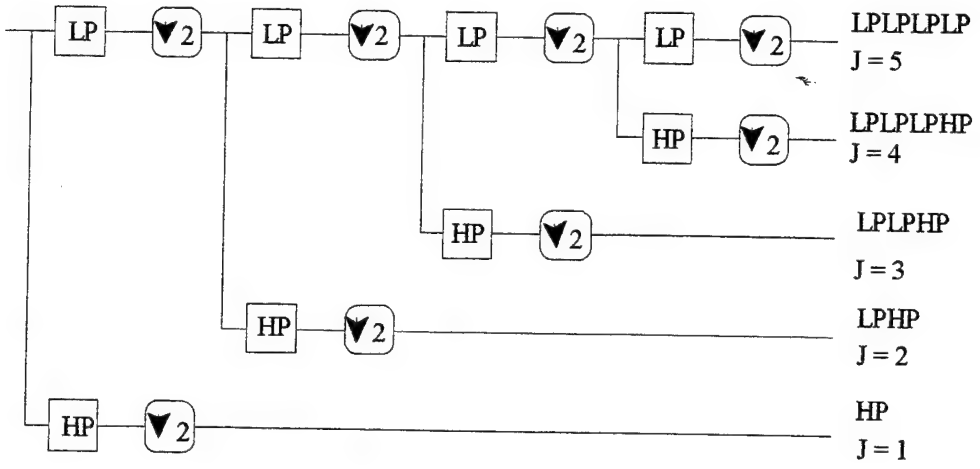

Figure 3.7: DWT implementation using filtering and down sampling operations.

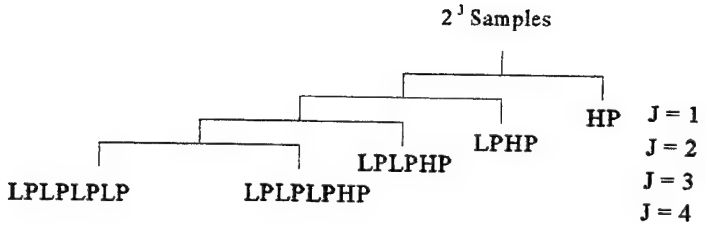

Figure 3.8: DWT tree structure.

A number of proposals have been made to deal with the time - variant nature of the wavelet transform. One method processes multiple time shifted versions of the input and averages the results (this is called "cycle spinning") [16]. Another method calculates all possible circulant shifts of the input signal using a fast algorithm developed by Beylkin [17], and averages the results. This has been shown to be equivalent to the undecimated DWT [16], and is a non-orthogonal transformation. A third approach is to seek an optimal shift of the input signal. In this case the transform becomes shift invariant, and orthogonal, but is signal dependent, since the shift is only optimal for the signal under consideration [15]. Some of these techniques have been applied to enhance signal denoising, and will be discussed again in Chapter VI.

4. Signal Reconstruction

Reconstruction of the original signal from the wavelet coefficients $C_{J,k}$ is achieved by reversing the quadrature mirror filtering and down sampling operations. The reconstruction will be exact however only if the QMFs (i.e., the wavelets) are properly chosen and exhibit some restrictive mathematical properties [13]. Such *perfect reconstruction filters* do exist, and are constructed by designing a second pair of QMF's that perform the interpolation (upsampling) and filtering operations. These *synthesis* filters entirely compensate for any amplitude, phase, and aliasing distortion of the *analysing* filter QMF's. Together, the analysis and synthesis filters form a two channel QMF bank, shown in Figure 3.9. The result is that the two channel QMF bank behaves like a linear, time-invariant system.[12]

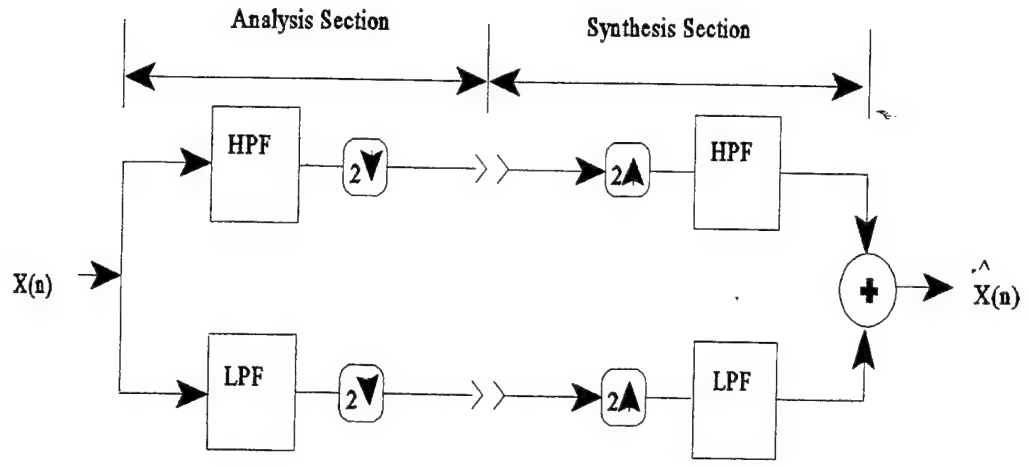

Figure 3.9: Two channel quadrature mirror filter. After Ref. [12].

C. WAVELET PACKETS

Reviewing the wavelet transform decomposition tree of Figure 3.8, it can be seen that the successive iterations of the LP part of the data produces a one-sided tree structure. The remaining tree branches of the complete binary tree can be produced by passing the HP part of the data through the QMF at each step as was done for the LP part. This will subdivide the upper half of the frequency range as well as the lower half, thus adding branches to both sides of the tree at each filtering iteration. The resulting complete binary tree structure is shown in Figure 3.10.

Each filtering and decimation iteration represents an increase in the transform scale and produces a new level of the tree. Since each decimation operation reduces the data set by a factor of two, there are a maximum of J levels for a data set of length $N = 2^J$ samples. The number of branches will double at each iteration as we move down the tree producing 2^J branches at the bottom of the tree.

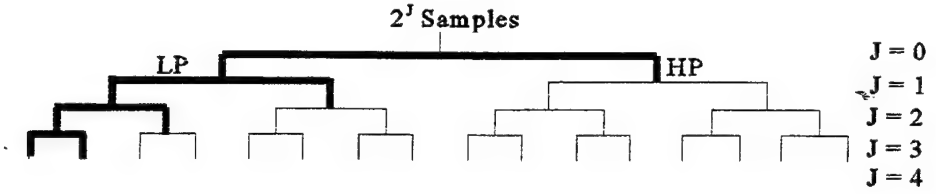

Figure 3.10: Complete wavelet packet decomposition tree structure.

The full binary tree clearly contains much redundant information since the information in any parent branch can be equivalently be replaced by the information of its two children branches at the next lower level [19]. Any complete level of the tree will form a complete orthogonal basis decomposition of the data. Additionally, by replacing a given parent branch by its children, a total of N x J possible decompositions can be produced. Any of these presentations will form a complete orthogonal basis and is considered a wavelet packet decomposition, consisting of a complete set of N wavelet packet coefficients. Figure 3.10 shows the wavelet basis decomposition in dark lines. Note that it is merely one of the many possible wavelet packet decompositions. In Figure 3.11 another decomposition is shown. Its corresponding tile diagram showing the division of the time-frequency plane is shown in Figure 3.12. These last two figures depict the opposite decomposition from the wavelet transform decomposition (shown in Figure 3.5). For this decomposition the time resolution is best at low frequency, and the frequency resolution is best at high frequency.

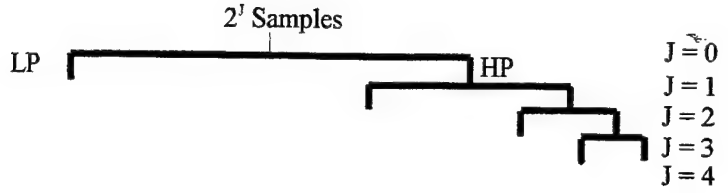

Figure 3.11: One possible wavelet packet decomposition.

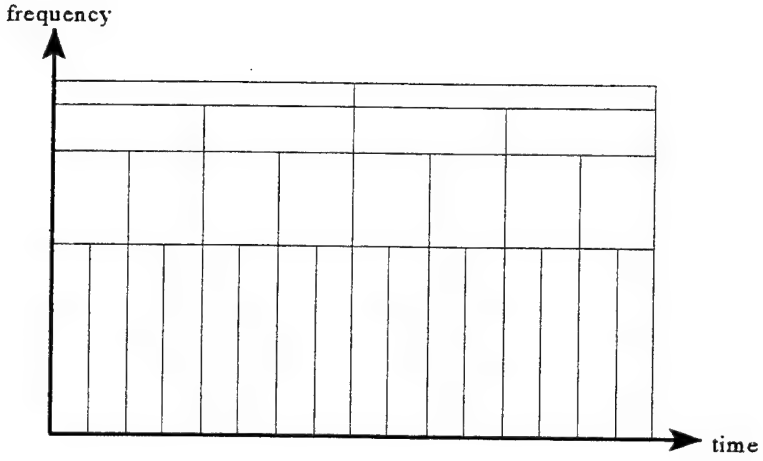

Figure 3.12: Tile diagram corresponding to wavelet packet decomposition of Figure 3.11.

IV. COSINE PACKET TRANSFORM

A. INTRODUCTION

The wavelet packet transform (WPT) provides a multi-resolution decomposition of the signal by smoothly partitioning the frequency axis [20]. The cosine packet transform (CPT) provides a multi-resolution capability by smoothly partitioning the time axis. The CPT therefore provides a complementary signal analysis tool to the WPT that is more suitable to certain types of signal decompositions. The CPT uses a windowed cosine function (the local cosine) as its basis. As a result, it performs extremely well on narrowband signals and signals with harmonic content. The WPT performs well on broadband pulse type signals. Use of both of these transformations allows for the processing of a wide variety of acoustic transients.

Two of the basic properties of a wavelet were described to be that they have zero average value (oscillate) and that they be of finite duration. As such it is not hard to imagine a wavelet that is simply a finite duration sinusoid. This is the basic premise of the local trigonometric transform. The local trigonometric functions consist of limited duration sinusoids multiplied by smooth cutoff functions (windows). These cutoff functions permit the smooth partitioning of the time axis, and allow for the construction of orthonormal bases on each interval [13,20].

The remaining sections of this chapter outlines the CPT and its two building blocks, the local cosine transform (LCT) and the discrete cosine transform (DCT). The

approach is descriptive, leaving out much of the mathematical details that can be found in reference [13].

B. DISCRETE COSINE TRANSFORM IV

The discrete cosine transform-iv (DCT-IV) is closely related to the Fourier transform, and has as its basis function a discrete version of the blocked cosine given by:

$$\cos[\frac{\pi(k+1/2)n}{N}] \qquad n = 1,2,.....,N \ , \ k \ an \ integer. \qquad (4.1)$$

Figure 4.1 displays the blocked cosine for $N = 256$, $k = 2$. Note that this limited duration cosine has a period of $2N$, and that its half integer frequency makes the left edge looks like a cosine while the right edge looks like a sine. The DCT-IV transform of a sequence $x(n)$ of length N is defined as:

$$X_c(k) = \sqrt{\frac{2}{N}} \sum_n x(n) \cos[\frac{\pi(k+1/2)n}{N}] \ , \qquad (4.2)$$

which is seen to be the inner product of the sequence with the blocked cosine. The DCT-IV can be related to the discrete Fourier transform in the following way. If we define a new sequence x_{2N} by evenly extending $x(n)$ such that;

$$x_{2N} = \begin{cases} x(n) & for \ n = 1,..., N \\ x(2N-n-1) & for \ n = N+1,...2N \end{cases} \qquad (4.3)$$

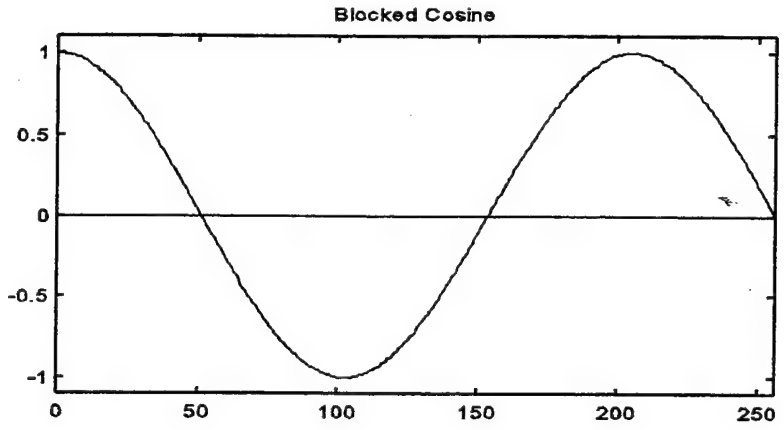

Figure 4.1: Blocked cosine.

then the discrete Fourier transform $X_F(k)$ of this new sequence x_{2N} is given by:

$$X_F(k) = \sum_n x_{2N}(n)\exp(\frac{-j2\pi kn}{2N}) . \qquad (4.4)$$

It can then be shown [21] that the DCT-IV transform of the original sequence is related to the DFT of the extended sequence by:

$$X_c(k) = \sqrt{\frac{2}{N}}\, X_F(k)\exp(\frac{-j\pi k}{2N}) . \qquad (4.5)$$

Equation 4.5 shows that the DCT of $x(n)$ can be computed from the DFT of the evenly extended sequence by multiplication by an additional matrix. This allows the DCT-IV to be computed quickly by taking advantage of speedy algorithms for finding the DFT.

C. LOCAL COSINE TRANSFORM

The blocked cosine of Figure 4.1 uses a rectangular window which causes distinct discontinuities at the boundaries and results in undesirable side-lobes in the frequency

domain. This problem can be overcome by the application of a smooth cutoff function that tappers the edges rapidly but smoothly at each boundary. The resulting function shown in Figure 4.2 is called a local cosine because of its localized nature in both the time and frequency domain. The smooth window or bell is typically constructed from a class of functions of the form $r(t) = e^{-jp(t)} \sin[\phi(t)]$. As an example, let $p(t) = 0$, and $\phi(t) = \pi/4[1+\sin(\pi t)]$ to form the bell given by:

$$r(t) = \begin{cases} \sin[\pi/4(1+\sin \pi t)] & -1/2 < t < 3/2 \\ 0 & \textit{otherwise} \end{cases} \qquad (4.6)$$

This bell $r(t)$ is shown in Figure 4.3 in both the time and frequency domain. The bell is symmetric about $t = 1/2$, and smoothly falls to zero at $t = -1/2$ and $t = 3/2$. Smooth partitioning of the time axis can be accomplished by overlapping bells, as shown in Figure 4.4, and the cosines will remain orthogonal despite this overlap [13].

In order to perform a local cosine transform of a data set the inner product of the data with the local cosine function is computed. However, the DCT-IV transform can be used by "folding" the overlapping portions of the bell back into the interval. The folding operation can be imparted on to the data, allowing direct application of the DCT-IV transform to a properly preprocessed (i.e., folded) data set. To reconstruct the signal the DCT-IV operation is applied to the transformed data, the result of which is then "unfolded" to produce the smooth overlapping segments. Details of the folding and unfolding operations can be found in references [13,22].

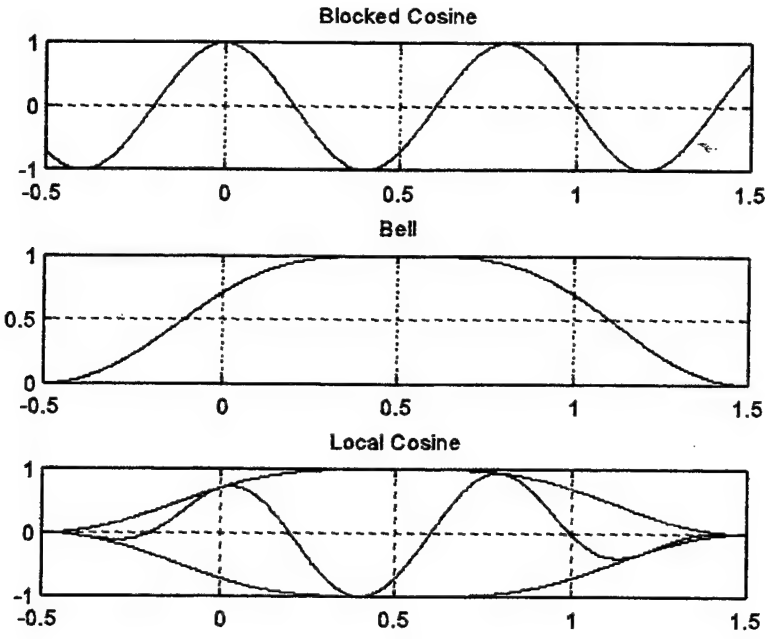

Figure 4.2: The local cosine.

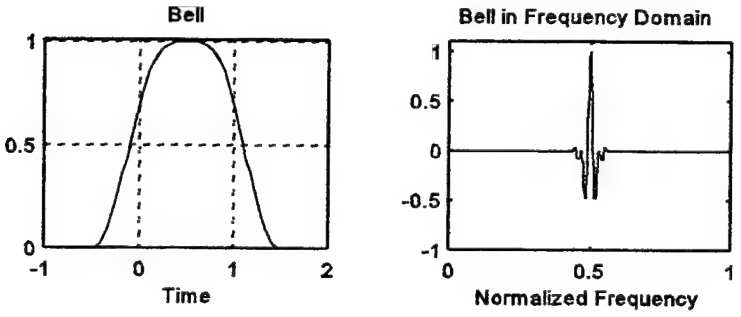

Figure 4.3: Bell function in time and frequency domain. Plot shows the real part of bell spectrum .

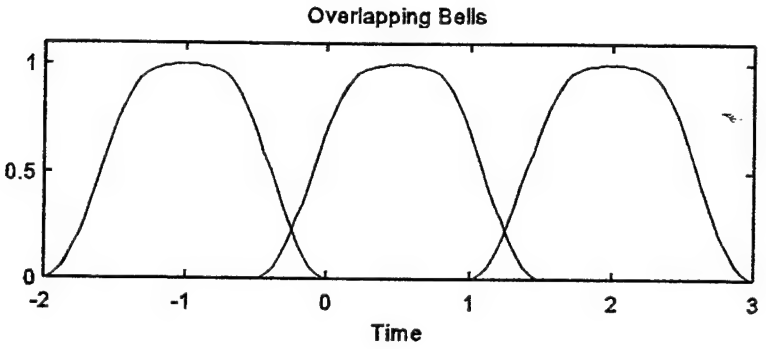

Figure 4.4: Overlapping bells in the time domain.

D. THE COSINE PACKET TRANSFORM

The local cosine permits partitioning of the time axis into arbitrary segments. The cosine packet transform provides a time division technique that segments the signal in a natural way. The signal is divided at its midpoint into two equal length time blocks. Each of these blocks is again divided at their respective midpoints. The splitting continues recursively until the blocks contain two samples. The result is a binary tree which looks similar to that of the wavelet transform decomposition, however the divisions in this binary tree are in time not frequency (scale). The cosine packet decomposition is depicted in Figure 4.5.

Once the signal has been partitioned in time it is transformed into the frequency domain by applying the local cosine transform using the fast DCT-IV algorithm. Since each time segment is windowed by the local cosine bell functions, orthogonal bases can be constructed using any combination of segments that cover the entire interval. Thus, similar

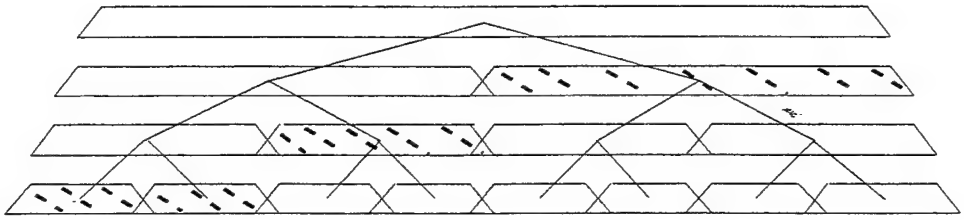

Figure 4.5: Cosine packet decomposition tree.

to the wavelet packet tree structure, any complete level constitutes an orthogonal basis, and any parent node can be equivalently replaced by its two children at the next lower level. The result is a total of $N \log_2(N)$ possible orthogonal bases, each of which is consider a cosine packet.

The CPT provides a binary tree structure where, at each successively lower time level, the time resolution improves and the frequency resolution decays by a factor of two. This is the opposite effect obtained from the WPT. The division of the time-frequency plane is displayed in the tile diagram of Figure 4.6, which is shown for the case corresponding to the highlighted blocks of Figure 4.5.

The CPT and WPT provide complementary methods for decomposing a given signal into the time-frequency plane. Each of these transforms result in a highly redundant set of coefficients, allowing for numerous presentations (ie., orthogonal bases) of the signal in the corresponding transform domain. Selecting the "best" presentation for a given application is the subject of Chapter V.

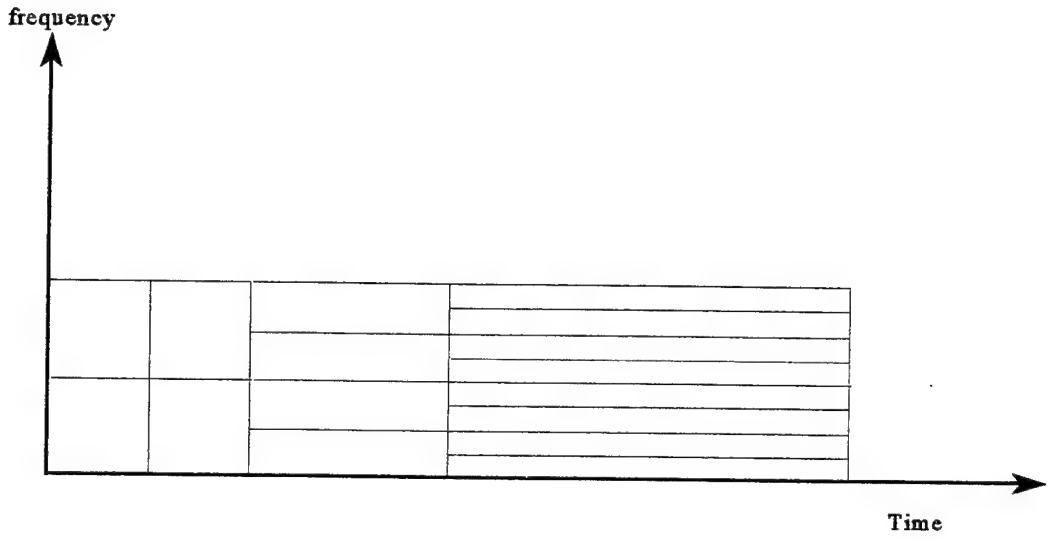

Figure 4.6: Tile diagram corresponding to Figure 4.5.

V. SELECTING A BASIS

A. INFORMATION COST

Given the large number of basis presentations available in a typical wavelet or cosine packet decomposition, selection of the "best" basis requires a comparison of the possible choices. The criterion for comparison will depend on the application and objective of the user. The method of noise removal employed in this study depends on the ability to transform the data into a relatively few large coefficients representing the signal and smaller coefficients representing the noise. For this application the best basis is the one that most compactly represents the data by concentrating the signal information into the fewest significant coefficients. This concept can be associated with an information cost function Q.

Information cost functions measure the expense (cost) of storing or transmitting a given sequence $x = \{ x_i \}$ [7]. These functions can be defined in many ways but to be useful Q must be additive such that $Q(0) = 0$, and $Q(x) = \sum Q(\{x_i\})$. This additive property simply means that the information cost of the sequence is the sum of the cost of its elements. Note also that Q is independent of rearrangement of the sequence elements. Two examples of cost functions are;

1. Count the number of elements in a sequence that exceed a arbitrary threshold.
2. Compute the l^2 norm or energy of the sequence.

The first would provide a measure of the cost of storing or transmitting the information necessary to reconstruct the original sequence to an arbitrary precision. The second would provide a measure of the total energy contained in the sequence.

Another cost function can be defined as:

$$Q(x) = \sum_i |x_i|^2 \log(1/|x_i|^2), \qquad (5.1)$$

which is known as the Shannon entropy [23]. This quantity is closely related to notions of probabilistic uncertainty and information of a sequence. This is the measure used in this thesis, and the following paragraphs summarize some of the properties associated with it.

In signal processing the information gained from observing a single element x_i of a signal $\dot{x} = \{x_i\}$ can be found from the expression:

$$I(x_i) = \log(1/p_i) , \quad I = 0 \ \ for \ \ p_i = 0, \qquad (5.2)$$

where $p_i = |x_i|/||x||^2$ is the normalized energy of the i^{th} element of the signal. The quantity p_i is a probability distribution function in that, $0 \le p_i \le 1$ and $\sum p_i = 1$. The entropy of the signal x_i is then defined as the expected value (mean) of $I(x)$ over the length of the signal and is given by:

$$H(x) = E[I(x_i)] = \sum_i p_i I(x_i) = \sum_i p_i \log(1/p_i). \qquad (5.3)$$

$H(x)$ is the entropy of the signal, and is a measure of the average information content per symbol of the sequence x. [7]

Two properties of $H(x)$ make it extremely useful in comparing two signals. The first is that it provides a measure of the concentration of the energy in the signal. For example, if two signals contain equal energy but different entropy, the signal with the lower entropy has its energy concentrated in fewer elements. The second property is that entropy allows the measurement of the rate of decay of a decreasing sequence, with the result that faster decaying sequences have lower entropy. [13]

These properties of $H(x)$ make it ideal for comparing two or more signal decompositions for their ability to concentrate the signal information into a relatively few coefficients. Note that $H(x)$ does not possess the additive property since $H(x) \neq \sum H(x_i)$, however $Q(x)$ of equation 5.1 is additive, and it is related to $H(x)$ by ;

$H(x) = \|x\|^2 Q(x) + \log(\|x\|^2)$. So that by minimizing the Shannon entropy $Q(x)$, the entropy $H(x)$ is also minimized.

B. THE BEST BASIS ALGORITHM

The Best Basis algorithm developed by Wickerhauser and Coifman provides a rapid way to analyze the numerous basis choices according to an information cost criterion [24]. In brief, the Best Basis algorithm works as follows:

1. Calculate and assign an information cost value to the coefficients in each node of the binary tree decomposition.
2. Compare the cost associated with each parent node to the sum of its two children nodes and flag the lower of the parent or the children.

3. Search the binary tree from top down and assemble the lowest cost flagged nodes into a basis.

Figure 5.1 graphically describes the process. In Figure 5.1a the information costs have been computed and assigned to each node. Figure 5.1b shows the results of comparing parent and children nodes with the lower cost value assigned to that of the parent (shown in parentheses). Any parent node with a information cost less than the sum of its children nodes is flagged (shown with an asterisk). Figure 5.1c shows the best basis resulting from searching the binary tree from top to bottom selecting the highest most asterisked nodes. Note the resulting best basis corresponds to the tile diagram of Figure 4.6.

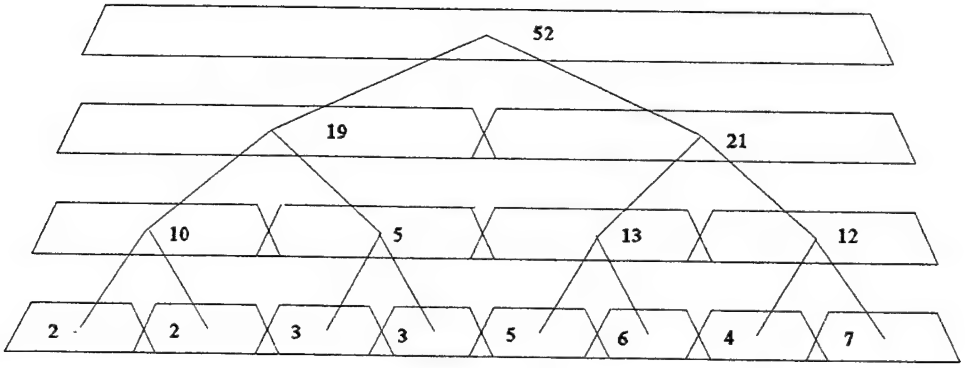

Figure 5.1a: Cosine packet decomposition tree with cost values assigned to each node.

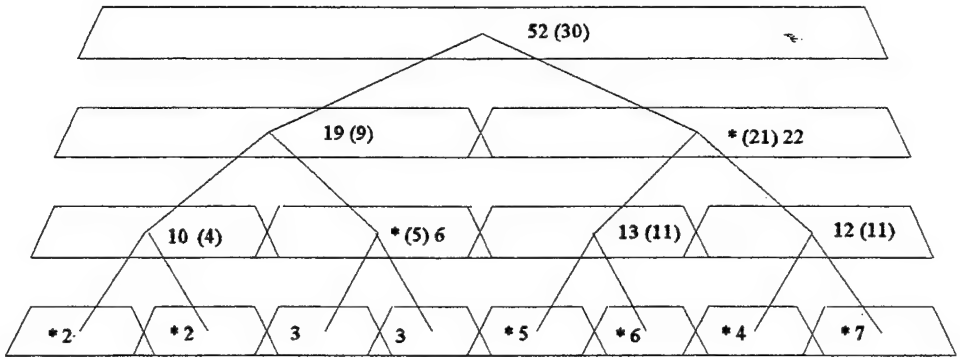

Figure 5.1b: Bottom up comparison of parent to children with lowest cost assigned to the parent node and shown in parentheses.

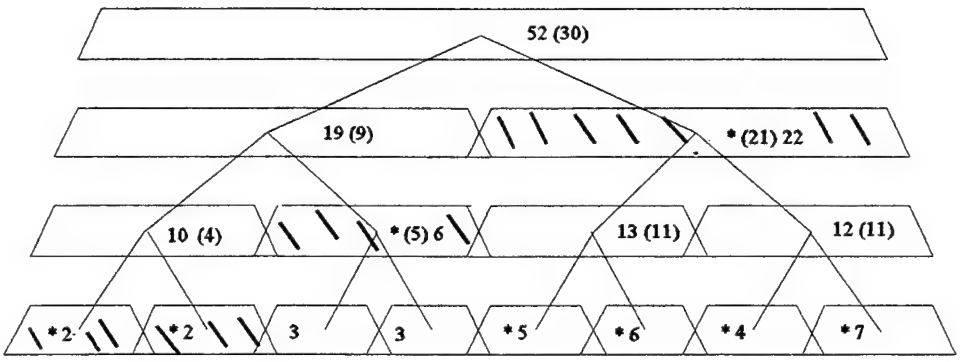

Figure 5.1c: Selection of best basis by choosing topmost asterisked nodes.

C. COMPARISON OF TWO SIGNALS

This section provides examples of the application of the information cost criterion (Shannon entropy) and the Best Basis algorithm. The first signal to be investigated in this application is an impulse function shown in Figure 5.2a. This impulse function was expanded into both the wavelet packet (using the Symmlet 8 wavelet) and cosine packet decompositions. The Best Basis of each transform was chosen among these expansions using the Best Basis algorithm, and is displayed as a tree diagram in Figure 5.2b. From Figure 5.2b, it can be seen that the best WPT basis consists of the coefficients from the second level of its decomposition tree and produces essentially zero for the information cost. The best CPT basis requires an ensemble of numerous tree branches to represent the impulse signal and has a resulting information cost of approximately 2.23. Figure 5.2c displays the associate wavelet packet and cosine packet normalized coefficients arranged in magnitude order. From this figure it is clear that the CPT requires many more coefficients to represent the impulse.

The second example signal is a cosine constructed of 2048 points, and shown in Figure 5.3a. This signal was likewise decomposed using both the WPT (with the Symmlet 8 wavelet) and the CPT. The resulting Best Basis tree structures, and sorted coefficients are shown in Figures 5.3b, and 5.3c respectively. In this case it is clear that the CPT provides the more efficient representation of the signal, as indicated by the lower Shannon entropy and the smaller total number of coefficients in the decomposition.

These two simple examples show that the Best Basis algorithm is capable of selecting the lowest cost basis from among several choices, and that the Shannon entropy provides a reasonable measure to compare the efficiency of bases to represent a signal. Additionally it points to the strengths and weaknesses of the CPT and WPT on certain types of signals. Short duration, broadband pulse transients (like that of the single impulse function) are best decomposed using the WPT with short duration wavelets like the Symmlet wavelet. Harmonic signals (like the cosine function) are best decomposed using the CPT because of its sinusoidal basis function.

Wavelet selection plays an important role in the results obtained by wavelet analysis. Although a number of wavelets were considered, the Symmlet 8 wavelet was chosen, and is used in all comparison testing of various techniques on this thesis. The Symmlet 8 wavelet was selected based on its robust performance on a wide variety of signals types, and its particularly good performance on the short broadband transients considered in this study.

The proceeding chapters have provided an introduction to wavelet analysis by discussing three transforms, the DWT, WPT, and CPT. Chapter VI will discuss how these transforms can be applied to remove noise from an input signal.

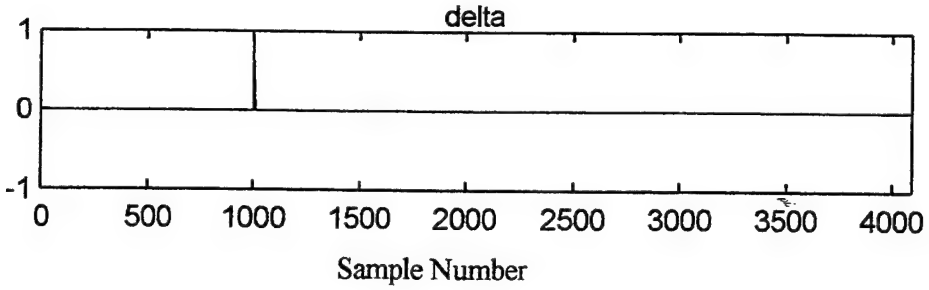

Figure 5.2a: Impulse function used to compare WPT and CPT bases.

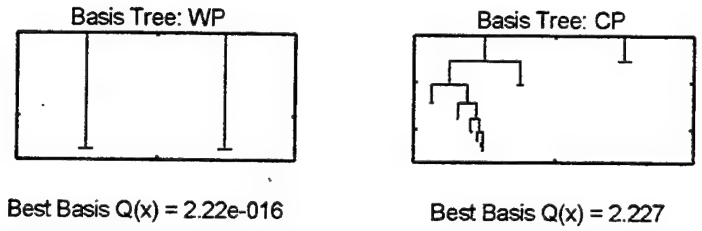

Figure 5.2b: WPT and CPT Best Basis tree diagrams obtained for the impulse function of Figure 5.2a.

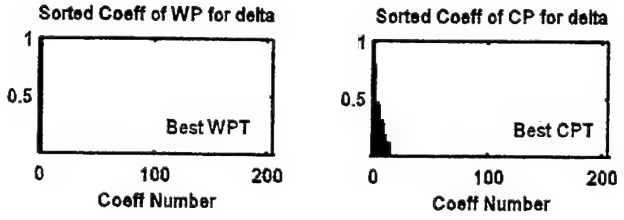

Figure 5.2c: WPT and CPT sorted coefficients obtained for the impulse function of Figure 5.2a.

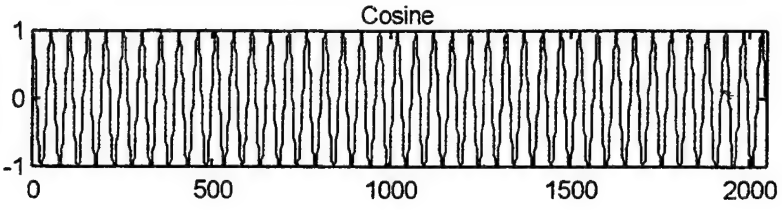

Figure 5.3a: Cosine function.

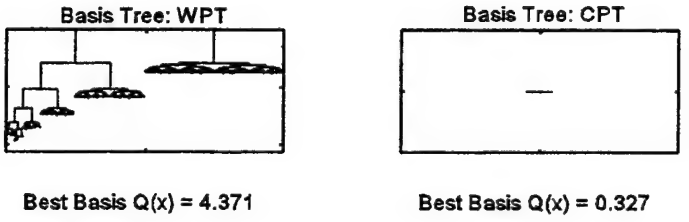

Figure 5.3b: WPT and CPT Best Basis tree diagrams for the cosine function shown in Figure 5.3a.

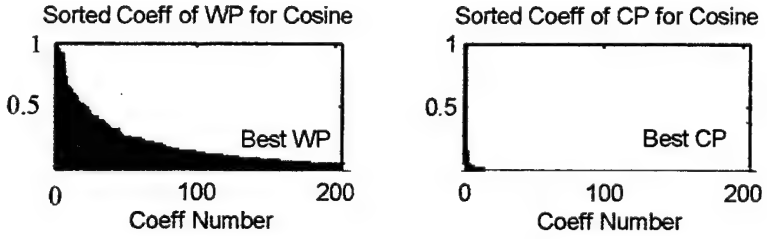

Figure 5.3c: WPT and CPT sorted coefficients obtained for the cosine function shown in Figure 5.3a.

VI. WAVELET-BASED DENOISING

A. PRINCIPLES

Decomposition of the data by the DWT, WPT or CPT, results in a matrix of coefficients that represents the data in the corresponding transform domain. This matrix contains all the information necessary to reconstruct the original input from the corresponding component wavelets or local trigonometric functions. The large coefficients represent good correlations of the input with the decomposing basis, conversely the small coefficients represent poor correlations of the input with the decomposing basis function. If the input was reconstructed neglecting some of the smaller coefficients, the reconstruction would still maintain the general shape of the original. However, there would be inevitable distortion introduced from simply neglecting some of the components necessary for the perfect reconstruction.

The idea in denoising is to judicially choose which coefficients to retain in order to preserve the signal while removing those that represent the noise. Two properties of the wavelet-based transforms assist in separating the noise coefficients from the rest. The first property is that, by properly choosing the basis to match the signal characteristics, the resulting decomposition will have a low information cost and will contain relatively few significant coefficients. The second property is that, for an input sequence that is a zero mean random process with uncorrelated samples (white noise), the transform coefficients will remain uncorrelated. If the the input sequence is additionally Gaussian distributed, the

coefficients will be Gaussian distributed and independent [23]. In this sense, the orthogonal wavelet-based transforms are linear operations which will transform white noise into white noise [25].

Therefore, the addition of noise to an input sequence will produce noisy coefficients, with the noise contributing to all coefficients, but the signal contributing to relatively few. In other words, for a suitably chosen basis applied to decompose a noisy input, good correlations will be produced with the signal (resulting in a few large coefficients), and poor correlations will be produced with the noise (resulting in many small coefficients). Observation of these properties leads to the idea of establishing a cutoff level (threshold) for those coefficients to be retained.

The general denoising procedure can thus be summarized as follows;

1. Decompose the input into a suitable basis using wavelet-based transforms.
2. Suppress the noisy coefficients by applying a non-linear thresholding method.
3. Reconstruct the signal using the inverse transform.

B. CALCULATING A THRESHOLD VALUE

1. Estimating the Noise

The term threshold refers to a number that is computed as a cutoff value to separate the coefficients that will be retained from those that will be suppressed or modified. The general methodology for calculating a threshold is based on the statistical properties of the transformed coefficients. If the coefficients are viewed as a series of

noisy observations $y(n)$, then the following parameter estimation problem can be formulated;

$$y(n) = f(n) + \sigma z(n) , \qquad n = 1,2,... \qquad (6.1)$$

where $f(n)$ is the deterministic function to be estimated, $z(n)$ is an i.i.d. N(0,1) random variable, and σ is the standard deviation of the noise. The objective is to suppress the noise and to recover $f(n)$ with the smallest mean square error.[25]

Any solution to this problem will have to assume or compute a value of σ. The standard procedure estimates σ as the *absolute median deviation* E of the coefficients at the finest scale divided by 0.6745 [25]. To explain this result, consider a random variable $X = \{x_i\}$ which is i.i.d. N(0,σ) and define E as [26];

$$E = med|x_i - med(X)| = med|x_i| . \qquad (6.2)$$

The operator *med* is the median, and the second equality in equation 6.2 results from the definition of X. Next, define q_1 and q_2 as two values of X that bound the center 50% of the distribution $X=\{x_i\}$. Figure 6.1 depicts the situation. From the standard normal distribution tables $q_2 = -q_1 = 0.6745 \cdot \sigma$. The absolute value of X will have 50% of its values bounded by $0 \le X \le q_2$, so that $med|x_i| = q_2 = E = 0.6745 \cdot \sigma$, or $\sigma = E/0.6745$.

This method of estimation of the noise standard deviation σ is robust because the transform coefficients at the finest scale will be essentially due to the noise, and any small number of coefficients due to the signal will not grossly effect the median.

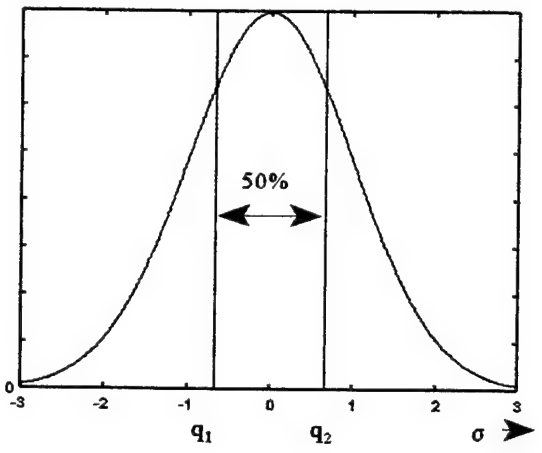

Figure 6.1: Normal distribution curve indicating center 50%.

Once the noise level of the transformed data is estimated a threshold value can be set. The simplest choice is to set the threshold (T) at some constant multiple of the noise standard deviation (e.g., $T = m \cdot \sigma$, where m typically lies in the interval $2 < m < 5$). Four methods of computing a threshold value are described below.

2. The Universal Threshold

The universal threshold value is based on a statistical theorem from reference [27] which states;

Given $X_1, X_2, ..., X_N$ where the X_i are i.i.d. N(0,σ), then as $N \rightarrow \infty$

$$P(\max_i |X_i| \leq T_u) = 1 , \qquad (6.3)$$

where T_u is given by;

$$T_u = \sigma \sqrt{2 \ln(N)} \ . \qquad (6.4)$$

Equation 6.3 indicates that, for a Gaussian distributed random variable X in the limit of large sample sizes N, no element X_i will have a magnitude greater than the quantity T_u. The quantity T_u is known as the universal threshold.

3. Steins Unbiased Risk Estimator (SURE) Threshold

This method of threshold calculation was proposed by Donoho and Johnstone [2], and is based on the work of Stein [28] in the area of multivariate normal distributions. This statistical procedure calculates the estimated mean square error (risk) for a range of threshold values, and selects that threshold value (T_S) with the resulting minimum risk.

4. Hybrid Threshold

The SURE threshold T_s is known to provide inaccurate results in the case of low signal energy [2]. In these cases the threshold estimate is biased unfavorably by the dominating noise coefficients and produces a faulty threshold value. The hybrid threshold (T_H) chooses between T_S and T_u based on the signal energy detected. It will select T_S only if sufficient evidence exists that the signal is significant.

5. MiniMax Threshold

The minimax principle is used to construct optimum estimators in the field of statistics. It is designed to select the choice of estimators that minimizes the worst case (maximum) errors of the set. Application of this method to wavelet thresholding was

proposed by Donoho and Johnstone [25], and expanded on by Bruce and Gao [29], where they tabulate the values of minimax thresholds T_M as a function of the sample size.

6. Comparison

Table 6.1 was constructed to compare the values of the thresholds achieved for different sample sizes of a zero mean white Gaussian noise signal N(0,1).

Sample size N	1024	2048	32768
T_U	3.723	3.905	4.560
T_S**	2.544	2.679	3.433
T_H	3.723	3.905	4.560
T_M	2.050	2.230	2.950

Table 6.1: Comparison of threshold values at different sample sizes.
** T_S is found statistically. The value shown is averaged using ten trials.

Note from Table 6.1 that the universal threshold value (T_u) is the largest, that the SURE (T_S) and minimax (T_M) threshold values are more conservative, and that the hybrid method (T_H) defaults to the universal value in this low signal energy case.

C. THRESHOLDING METHODS

Once a threshold value is established a number of methods exist to apply the threshold to suppress or modify the coefficients of the decomposition. Three different thresholding methods are considered in this study, hard, soft, and semi-soft.

1. Hard Thresholding

The non-linear hard thresholding function Λ^H is defined as:

$$\Lambda^H (C_{j,k}) = \begin{cases} C_{j,k} & |C_{j,k}| \geq T \\ 0 & |C_{j,k}| < T \end{cases}, \qquad (6.5)$$

where T is the threshold set by the user [16,29]. In hard thresholding all the transformed coefficients with magnitudes that exceed the set threshold value are retained, and all the others are set to zero. One difficulty with hard thresholding is that removal of all fine detail from the signal may produce fictitious oscillations and create contrast where none previously existed [25].

2. Soft Thresholding

The non-linear soft thresholding function Λ^S is defined as:

$$\Lambda^S (C_{j,k}) = \begin{cases} sign(C_{j,k})[|C_{j,k}| - T] & |C_{j,k}| \geq T \\ 0 & |C_{j,k}| < T \end{cases}. \qquad (6.6)$$

In soft thresholding all the transform coefficients with magnitudes smaller than the threshold value T are set to zero, and all the remaining coefficients are reduced in magnitude by the amount of the threshold value [25]. The advantage of this method is that the results are not as sensitive to the precise value of the threshold T selected, as in the "keep or kill strategy of hard thresholding. The disadvantage of this method is that the general shape of the signal might be slightly affected since the even the large coefficients are modified using this scheme.

3. Semi-Soft Thresholding

Semi-soft thresholding is a compromise between hard and soft thresholding methods. It was introduced by Bruce and Gao [29]. It uses two threshold values T_1 and T_2 where $T_2 > T_1$, and is defined as:

$$\Lambda^{SS}(C_{j,k}) = \begin{cases} C_{j,k} & |C_{j,k}| \geq T_2 \\ sign(C_{j,k}) \dfrac{T_2(|C_{j,k}| - T_1)}{T_2 - T_1} & T_1 < |C_{j,k}| < T_2 \\ 0 & |C_{j,k}| \leq T_1 \end{cases} . \qquad (6.7)$$

Coefficients with magnitudes between T_1 and T_2 are reduced, those with magnitudes above T_2 are retained, and the rest are set to zero. Note that for $T_1 = T_2$, this is hard thresholding.

D. COLORED NOISE

The calculation of the threshold value by the methods prescribed above is restricted to the case of signals in white Gaussian noise. Extension to colored noise environments was proposed by Johnstone and Silverman [30]. This method treats each scale of the transformed data as a Gaussian distribution. A threshold value is then calculated and applied to each scale. An alternate approach is to apply a pre-whitening transform to the data prior to its decomposition, which alleviates the need to calculate a separate threshold for every scale. The use of a pre-whitening transform was found to

produce better results than that of applying *level wise* thresholding on the data analyzed in this study.

The general denoising procedure chosen in this work can thus be summarized as follows;

1. Apply a pre-whitening transform to the input data.
2. Decompose the input into a suitable basis using wavelet-based transforms.
3. Suppress the noisy coefficients by applying a non-linear thresholding method.
4. Reconstruct the signal using the inverse transform.

E. A TEST CASE

1. Synthetic Data

In this section four synthetic signals are studied to understand the different thresholding methods, and assist in comparing results from one attempt to the next. These signals were chosen to capture some of the essential features of the real world acoustic transients of primary interest to us. The four test signals are called *Decay, Bumps, Doppler, and Heavisine*, and are shown in Figure 6.2. *Decay* is a decaying exponential, *Bumps* is a series of sharp peaks, *Doppler* is a down chirped sinusoid, and *Heavisine* is a distorted sinusoid with a discontinuity. All signals consist of 2048 data points. (The signals Bumps, Doppler, and Heavisine were created using the Wavelab .700 *makesignal.m* function [10]).

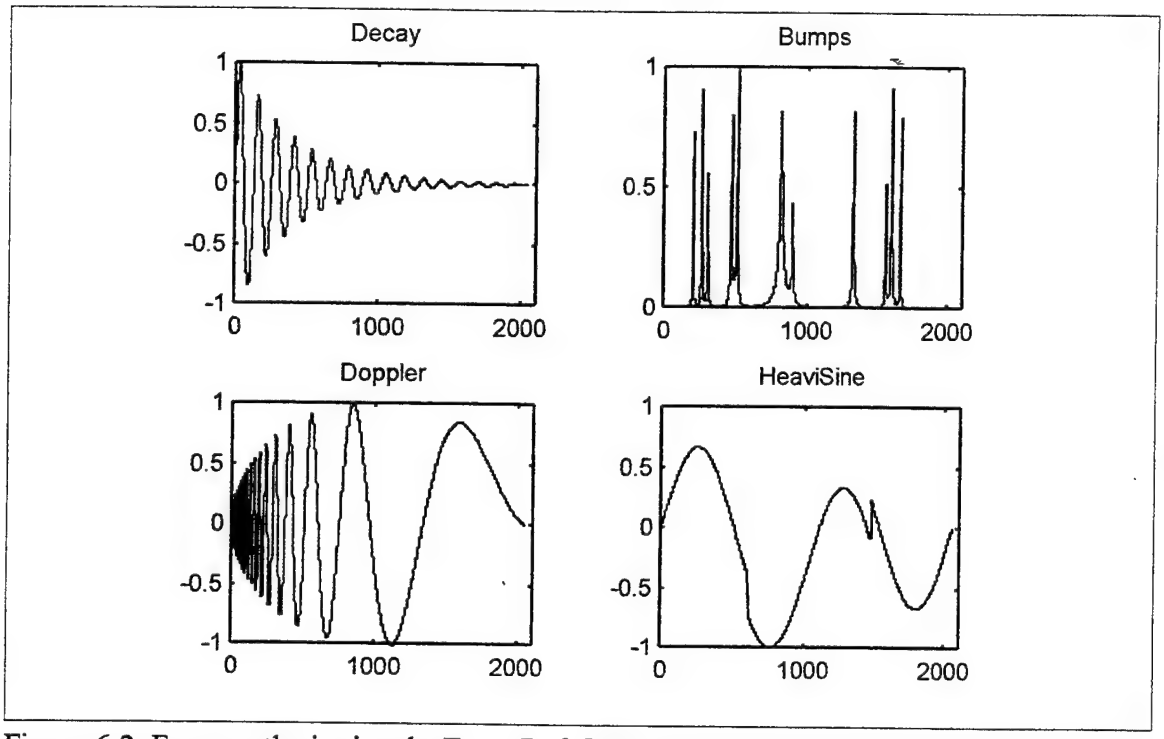

Figure 6.2: Four synthetic signals. From Ref. [10].

2. Comparison of Methods

The Wavelab.700 toolbox [10] was used to make an initial comparison of the performance of wavelets, wavelet packets and cosine packets on the four synthetic signals. The benchmark used to compare the cleaned signals is the Mean Squared Error (*MSE*), defined as:

$$MSE = \frac{1}{N}\sum_{n=1}^{N} [x(n)-y(n)]^2 \qquad n = 1,2,...,N, \qquad (6.8)$$

where $x(n)$ is the noise free original and $y(n)$ is the denoised output, and N is the length of the data.

a. Wavelet Transform

The first comparison was made by decomposing noisy versions of the synthetic signals into the wavelet basis by applying the discrete wavelet transform using the Symmlet 8 wavelet. The four synthetic signals were subjected to increasing amounts of white noise spanning the range of SNR's from +10 dB to -10 dB. The denoising was accomplished using the four different threshold values (T_u, T_S, T_H, T_M) and the hard and soft thresholding methods. The coefficients at the two highest scales (lowest levels) of the decomposition were exempted from the thresholding, based on the assumption that these coefficients primarily represent the signal. The resulting plots of *MSE* vs SNR are shown in Figures 6.3 and 6.4. Each plotted point represents the average *MSE* value obtained using ten trials at a given SNR level.

Figure 6.3 displays the application of the hard thresholding method. The universal threshold T_u produced the lowest *MSE* for all signals except at the low SNR values of the *Bumps* signal. Figure 6.4 displays the application of the soft threshold method and shows the minimax and SURE threshold values performing best on all except the *Heavisine* signal. These results suggest that the hard thresholding method performs better when used along with the larger threshold value T_u, and that the soft thresholding methods perform best using the more conservative threshold values of T_S and T_M. Also note that the four thresholding values performed nearly equally well when the soft thresholding method was applied, however the hard thresholding method shows more divergence between the choice of threshold values. This suggests that the selection of

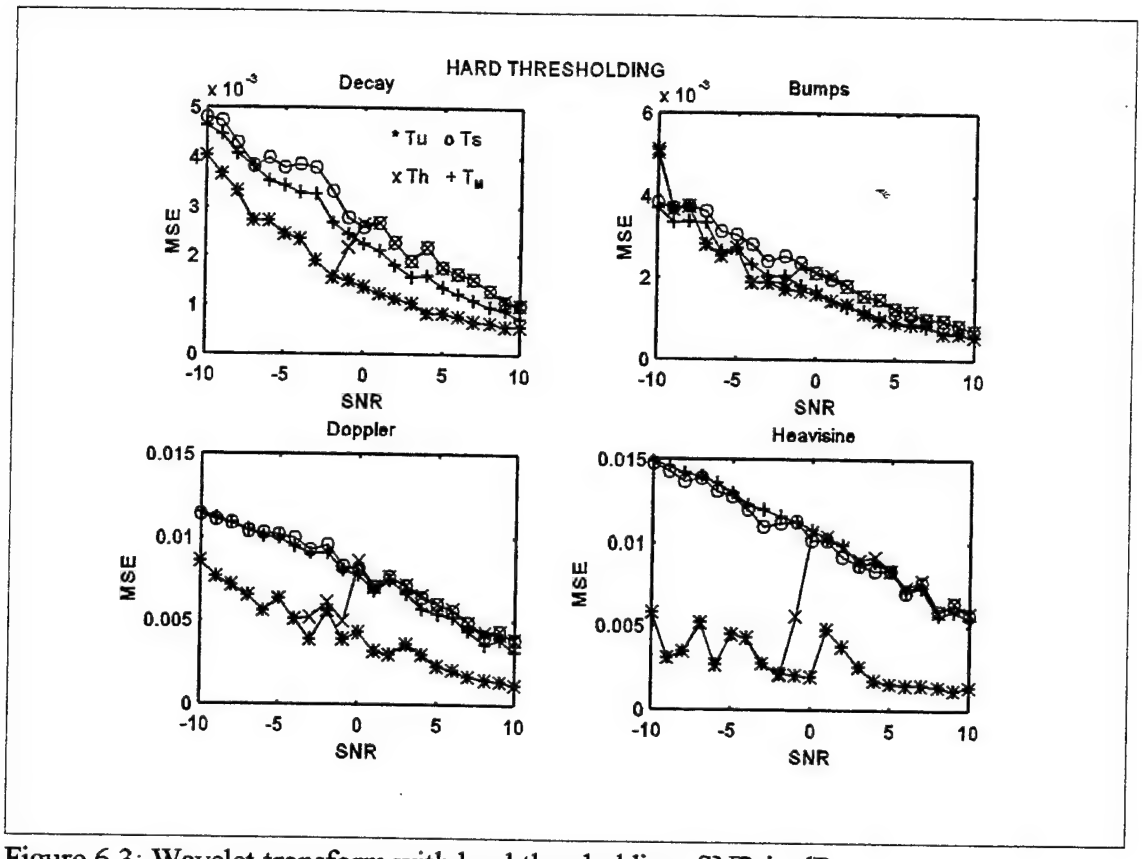

Figure 6.3: Wavelet transform with hard thresholding. SNR in dB.
** T_U : universal, oo T_S : Sure, ++ T_M : minimax, xx T_H : hybrid.

threshold value is more critical when employing the "keep or kill" strategy of hard thresholding.

b. Wavelet Packet and Cosine Packet Transforms

The second comparison was made by decomposing noisy versions of the synthetic signals using the WPT (using the Symmlet 8 wavelet) and the CPT. Each of these decompositions were denoised by application of the hard, soft, and semi-soft thresholding methods. The T_u threshold value was used for both the hard and soft thresholding, and the semi-soft thresholding used T_2=6.9, and T_1 = 2.8 as tabulated in reference [29]. The basis tree depth was limitted to eight levels, since experimentation

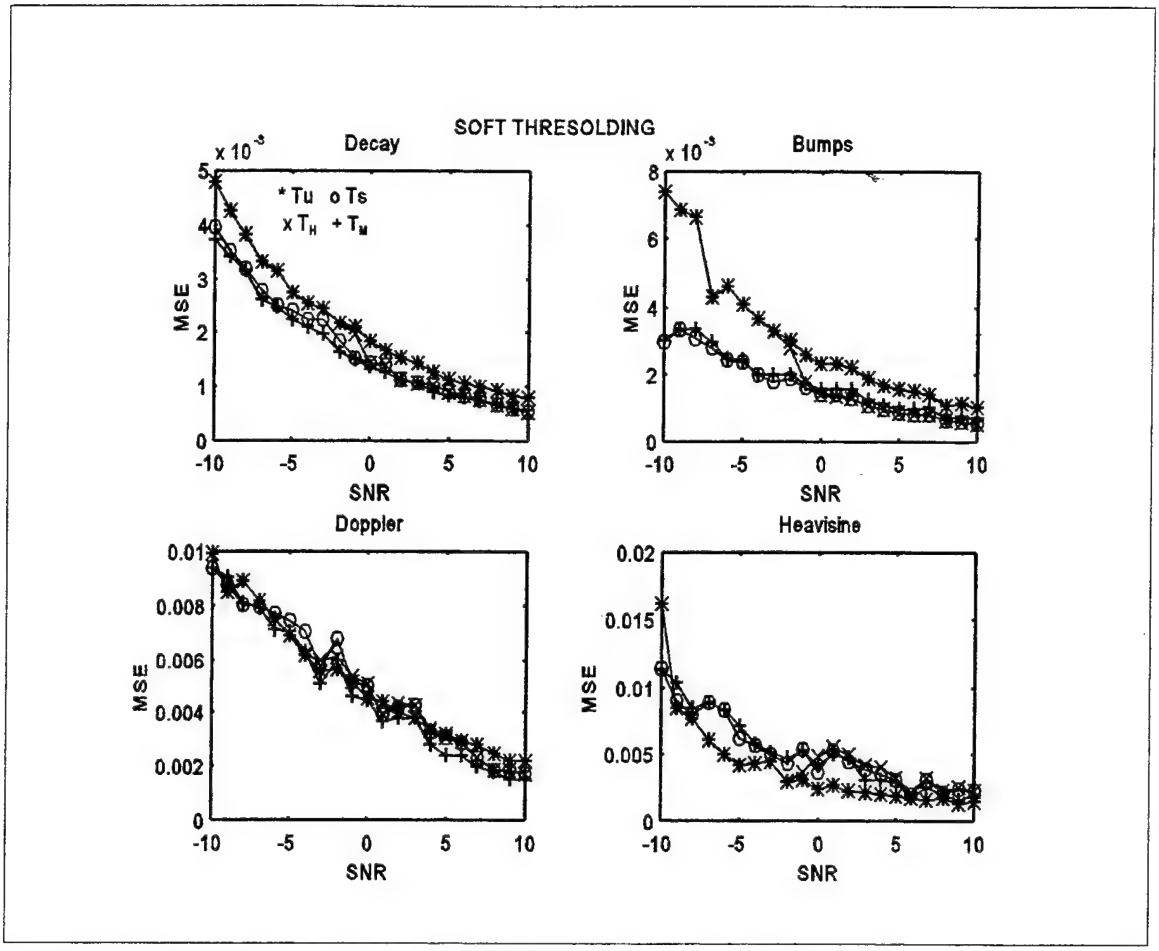

Figure 6.4: Wavelet transform with soft thresholding. SNR in dB.
** T_U : universal, oo T_S : Sure, ++ T_M : minimax, xx T_H : hybrid.

demonstrated that further decomposition did not improve the denoising.The resulting plots of *MSE* vs SNR are shown in Figures 6.5 and 6.6. Each plotted point represents the average *MSE* value obtained using ten trials at a given SNR level.

Overall the wavelet packet transform performed as well, or better than that of the wavelet transform. Such results are to be expected since the wavelet decomposition is a subset of the wavelet packet decomposition (wavelets are but one possible path through the wavelet packet binary tree). The cosine packet methods however also

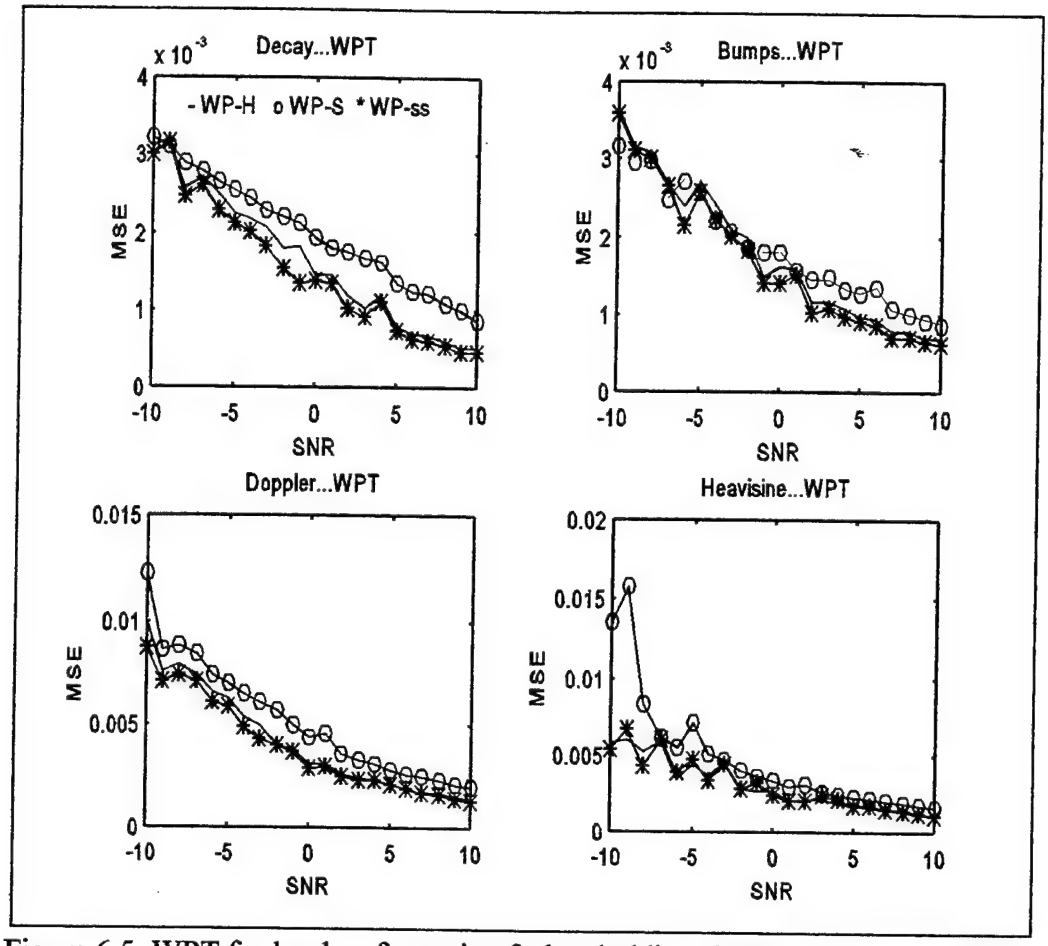

Figure 6.5: WPT for hard, soft, semi-soft thresholding. SNR in dB.
-- Hard, oo Soft, ** Semi-Soft thresholding.

performed well on the oscillating signals (Decay, Doppler, and Heaviside), and performed more poorly on the spiky Bumps signal. Also clear from the packet transform plots is that the wavelet packets using semi-soft threshold was consistently the best or nearly the best of all the methods, on all four signals. While the cosine packets with semi-soft threshold performed the best of the cosine packet methods on all four signals.

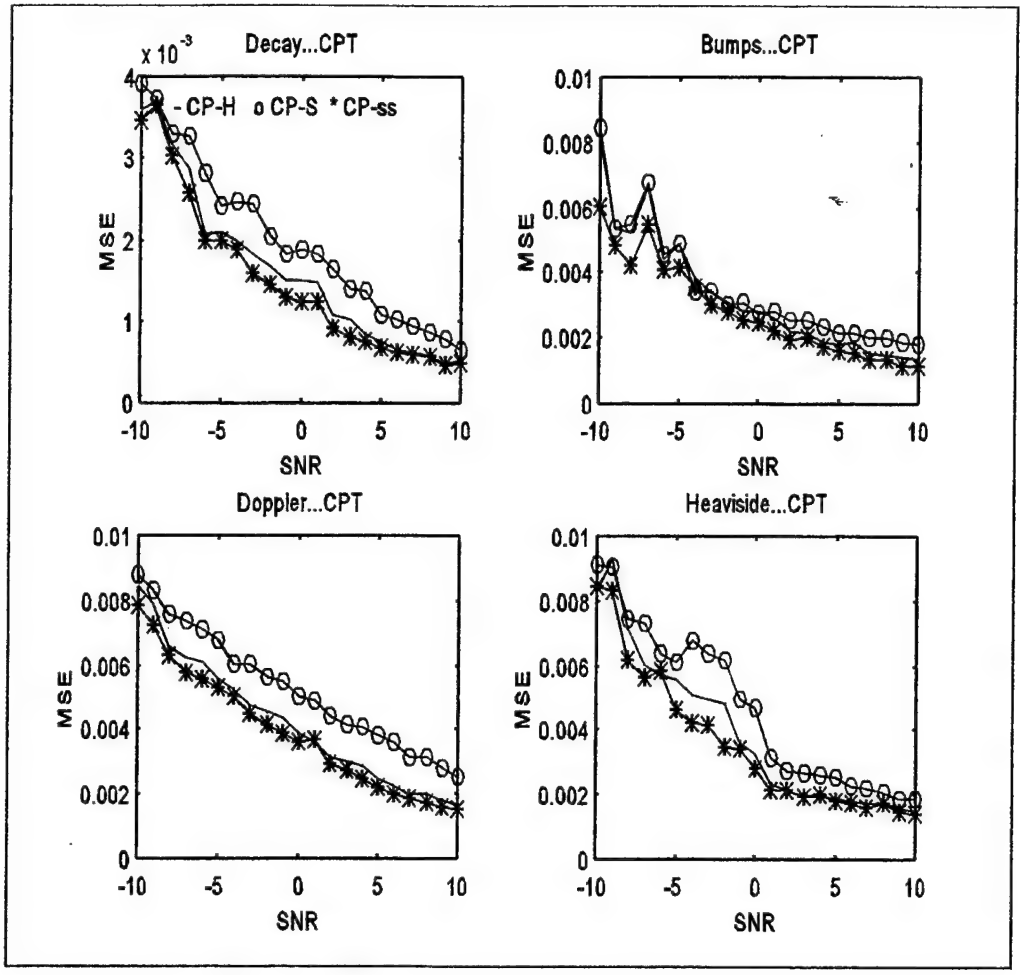

Figure 6.6: CPT for hard, soft, and semi-soft thresholding methods. SNR in dB. -- Hard, oo Soft, ** Semi-Soft thresholding.

c. Conclusions

Although the results could be interpreted a number of ways, a few general comment can be made. First, the packet transforms as a group perform better than the wavelet transform on the signals considered here. Second, the CPT performs best on oscillating (harmonic like) signals, while the WPT denoises spiky discontinuous signals

best. Finally, all thresholding denoising schemes lead to similar results as the SNR level increases.

Figure 6.7 shows the results obtained when denoising the signal *Decay* (at three different values of SNR), using two different thresholding schemes. The second column of this figure displays the results obtained when denoising with wavelet transform, with hard thresholding with the threshold value T_u. This case represents the best of the wavelet transform methods. The third column in Figure 6.7 displays denoising results obtained using the CPT, and semi-soft thresholding method. This case represents the best of the CPT methods. The CPT method provides a cleaner visual result and lower *MSE*. Such results are to be expected as the local cosine function is better matched to the *Decay* signal than the Symmlet 8 function is.

Figure 6.8 compares the results of hard, soft, and semi-soft thresholding methods (using the WPT) on the signal *Decay* at SNR= -2 dB. This figure displays the divergence between visual and numerical quality of the denoising. Although semi-soft thresholding achieves the lowest *MSE* it provides the worst visual reproduction of the clean signal. (Note that the slightly "step like" appearance of the plots is an anomaly introduced as the result of the graphical reproduction.)

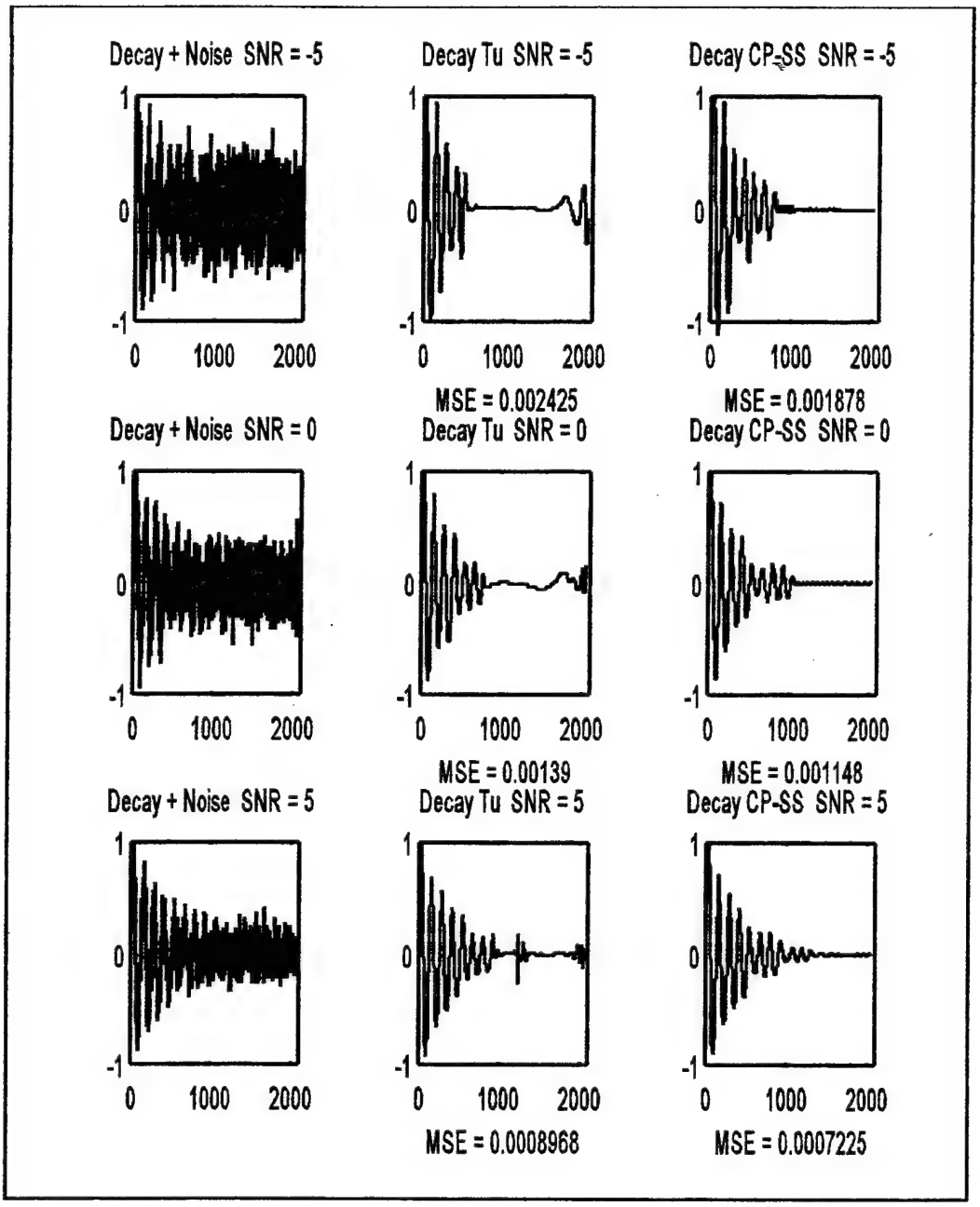

Figure 6.7: Denoising comparison of CPT using semi-soft thresholding and wavelet transform using hard thresholding.

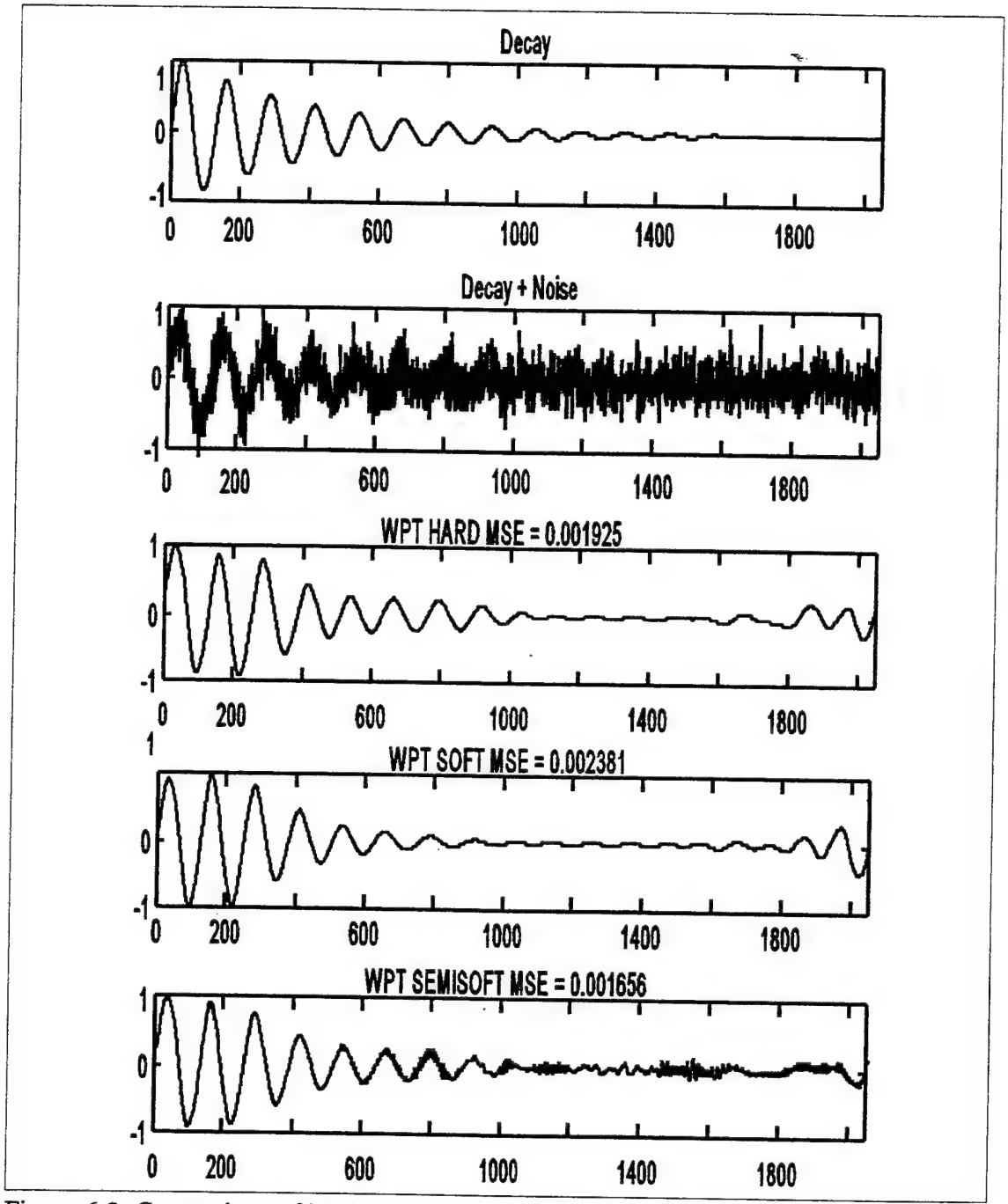

Figure 6.8: Comparison of hard, soft, and semi-soft thresholding. Denoising of signal *Decay* by the WPT using the Symmlet 8 wavelet. Note the disparity between *MSE* and visual quality.

F. TRANSLATION INVARIANT DENOISING

1. Cycle Spinning

Wavelet denoising with the orthonormal DWT sometimes results in the production of artifacts, which occur due to the unfortunate alignment of a signal discontinuity with that of the decomposing wavelet at a given shift and scale. These artifacts take the form of spurious oscillations in the neighborhood of the signal discontinuity [16]. An example of this behavior can be seen in the bottom center plot of Figure 6.7, where a short rapid oscillation was produced in the reconstruction of the denoised signal.

These artifacts are attributed to the lack of translation invariance of the DWT, since a signal with similar features but slightly different alignment in time or scale might exhibit fewer artifacts [16]. One possible solution is to manually shift the signal data to achieve a more favorable alignment. However, shifting the signal could improve the behavior near one discontinuity, while worsening the behavior near another signal discontinuity. Thus, an arbitrary shift of the signal data will not guarantee a consistently better result.

Coifman and Donoho [16], proposed a more formal procedure to suppress the artifacts called *cycle spinning*. This method "averages out" the translation dependence, by performing the denoising over a range of shifts of the input data and averaging the results. The technique is to shift the data, denoise, and then unshift the denoised data. Repeating this process for a range of shifts and then averaging the results has been shown to produce a reconstruction with significantly reduced artifacts.[16]

The cycle spinning technique is not limitted to use with the DWT, and can also be applied to the CPT and WPT as well. In fact, application of this method has been shown to reduce the "clicking" effect sometimes found at the data segmentation points of the CPT, and to reduce the isolated spikes sometimes produced from wavelet packet denoising.[16]

Figure 6.9 was produced using WaveLab .700 [10], and shows the improvement that can be obtained by using the cycle spinning technique. The top two plots display the signal *Decay* and its noisy version, respectfully. The third plot from the top displays of the results of denoising with the WPT using the Symmlet 8 wavelet. The bottom plot displays the result of denoising by averaging the results of eight shifted versions of the input data (i.e., eight spins) using the WPT and the Symmlet 8 wavelet. The cycle spinning method results in an improvement in both *MSE* and visual appearance.

2. Translation Invariant Discrete Wavelet Transform

For a data set of size N, computation of the DWT for *all* circular shifts can be computed in order $N \cdot log_2(N)$ time, and it is equivalent to computing the undecimated, non-orthogonal, discrete wavelet transform [18]. Denoising with this translation invariant discrete wavelet transform (TI-DWT) can provide improve performance over the orthogonal DWT, since it provides the averaging benefits described for cycle spinning.

The denoising procedure is to decompose the signal into the TI-DWT basis using the fast algorithm of reference [17], apply the same thresholding methods described earlier and then perform the inverse transform. Figure 6.10 was produced using Wavelab .700

[10], and shows the results of applying this technique compared to that of the DWT. The top two plots of the figure show the signal *Decay* and its noisy version. The third plot is the denoised output from the DWT (using the symmlet 8 wavelet) with hard thresholding, and the universal threshold value. The bottom plot shows the result of denoising with the TI-DWT using the same wavelet and thresholding parameters. The TI-DWT method thoroughly removes the artifact found in the DWT output, but does not provide an improvement in the MSE in this case.

During the application of these two techniques to the data in this study, we found that the TI-DWT generally provided better denoising results than that of the wavelet transform. However, use of the wavelet and cosine packet transforms for denoising performed better than both the TI-DWT and the wavelet transform. Additionally, both the CPT and WPT methods benefited from using the cycle spinning procedure with a small number of spins (e.g., eight spins).

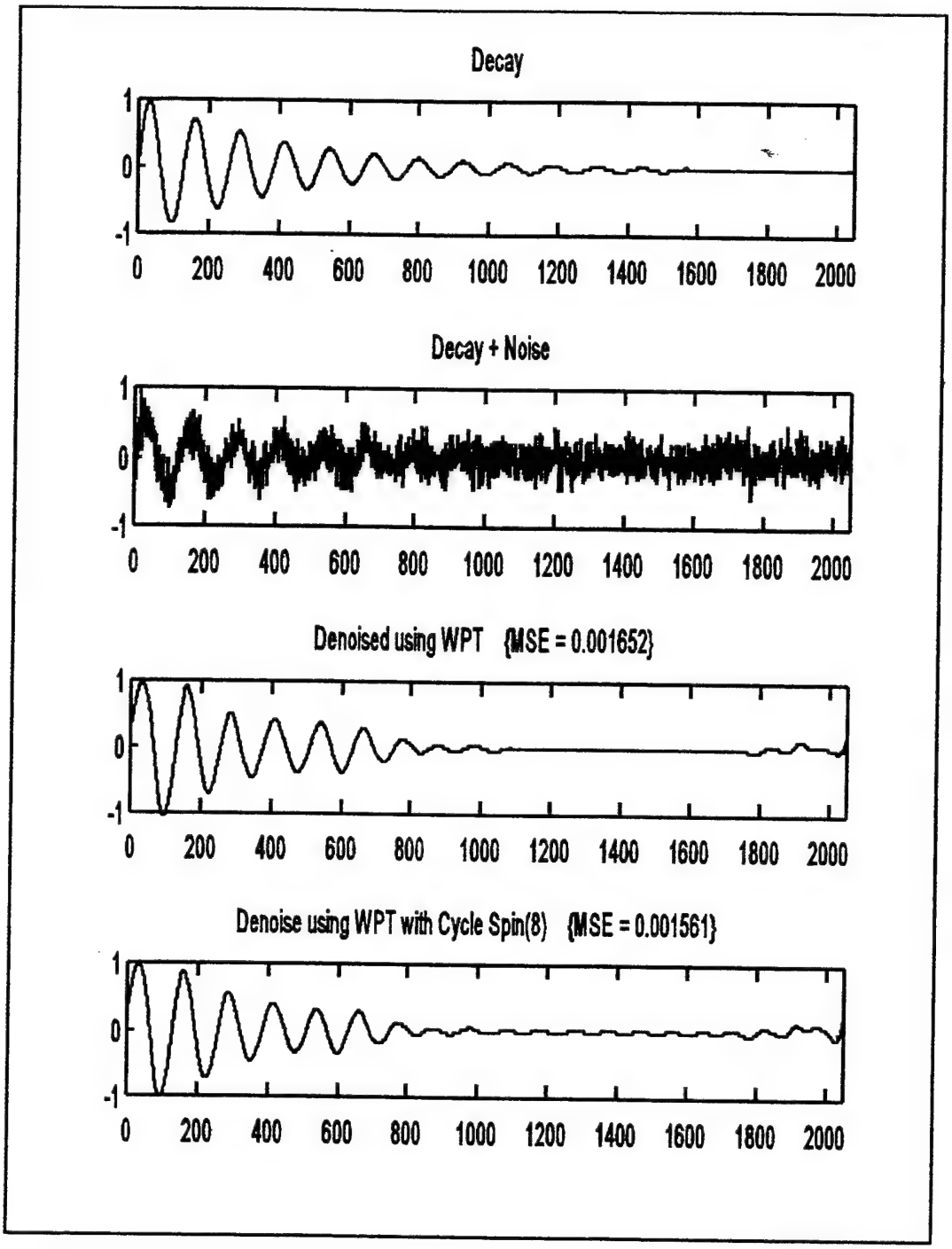

Figure 6.9: Comparison of Denoising with the WPT and WPT using Cycle Spinning. Top plot is original signal. Second plot is the noisy signal. Third plot is result of denoising with WPT (using Symmlet 8 wavelet). Fourth plot is the result of denoising using cycle spinning with eight spins.

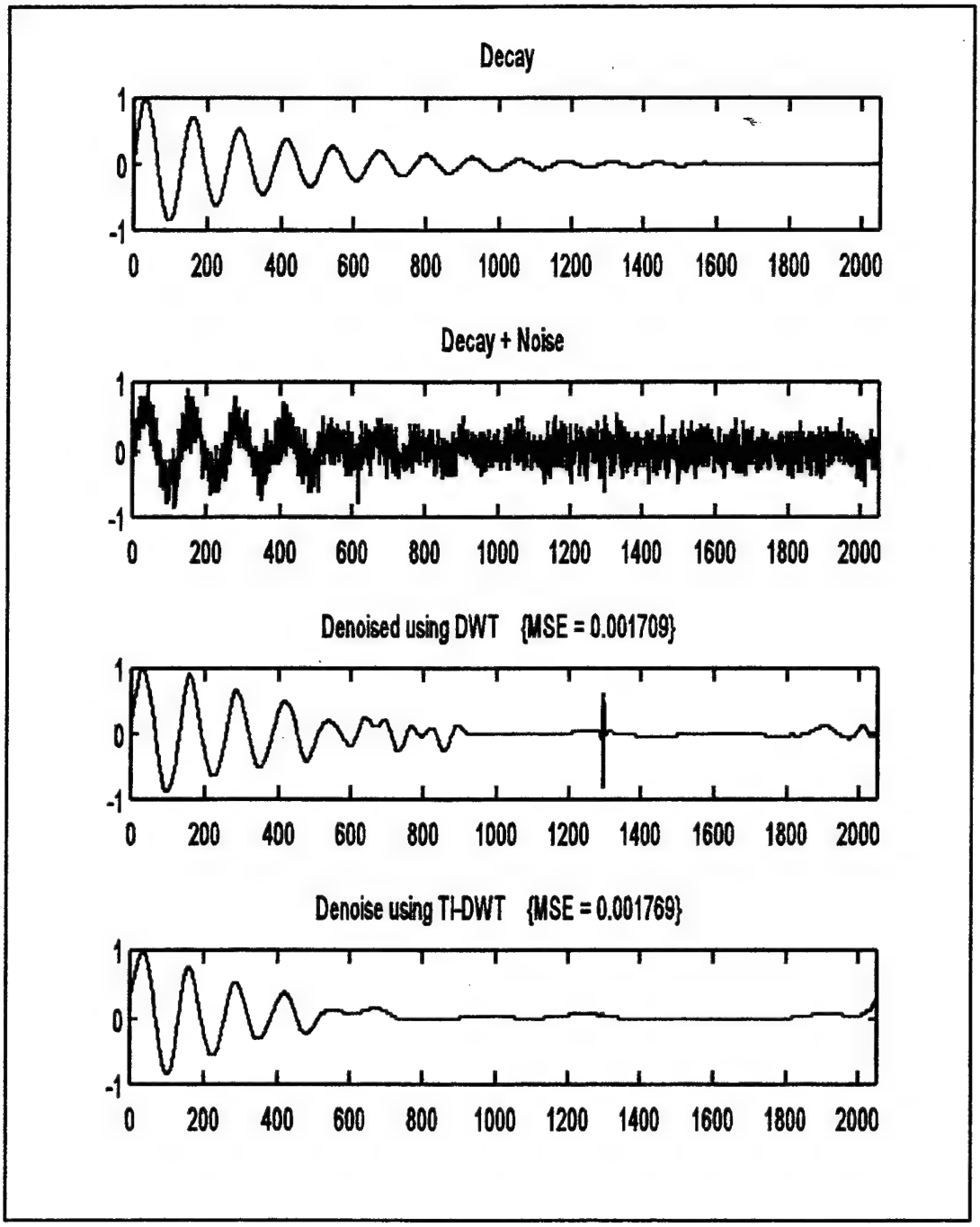

Figure 6.10: Comparison of denoising with DWT and TI-DWT.

VII. DENOISING ALGORITHM AND RESULTS

Chapter VI described and contrasted a number of wavelet based thresholding methods. This chapter will apply and extend those concepts to the removal of noise from underwater acoustic transients, and show the results achieved. Section A describes the details of the denoising procedure. Section B displays some results of applying the denoising procedure to real underwater data. Section C compares the performance of wavelet thresholding to that of a short-time Wiener filter.

A. DENOISING ALGORITHM

This section provides an algorithm for denoising acoustic signals by implementing the four step general denoising procedure outlined in Chapter VI.

1. Pre-whitening

Applying a pre-whitening transform to the input data permits extension of the thresholding rules to colored noise environments. The pre-whitening transformation is accomplished using the technique of Frack [31]. The general approach is to form an autoregressive (AR) model based on a sample of the input colored noise. Then the noisy data is filtered with a "whitening" filter that is formed from the inverse of the noise AR model. The output of the filter will be a colored version of the signal in additive white noise. The details of the AR model and whitening filter are given below.

a. Autoregressive Modeling

Any stationary random processes can be modeled as the output of a linear time-invariant filter that is subject to a white noise input [32]. The filter in an AR model is

an infinite impulse response (IIR) filter with a transfer function given by:

$$H_a(z) = \frac{1}{A(z)}, \qquad (7.1)$$

where

$$A(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots\cdots + a_p z^{-p}. \qquad (7.2)$$

The coefficients a_i of this all pole filter can be found by solving a system of linear equations relating the a_i 's to the correlation matrix of the random process to be modeled. The AR model parameters can also be derived directly from the data without the need to compute the correlation matrix or solve the system of linear equations. One clever technique for computing the parameters recursively is known as the *Burg Method* [32]. This is the method used by Frack [31], and has the advantage of guaranteeing a stable filter by ensuring that all poles of the model are kept within the unit circle. One disadvantage of the Burg Method is that it requires data lengths greater than a few thousand points to produce good estimates [31].

b. Prediction Error Filter

A linear predictor is designed to estimate the current or future values of a random sequence based on knowledge of the past sequence values. The output of a prediction error filter (PEF) is taken as the difference between the estimate of the linear predictor and the actual sequence, as shown in Figure 7.1.

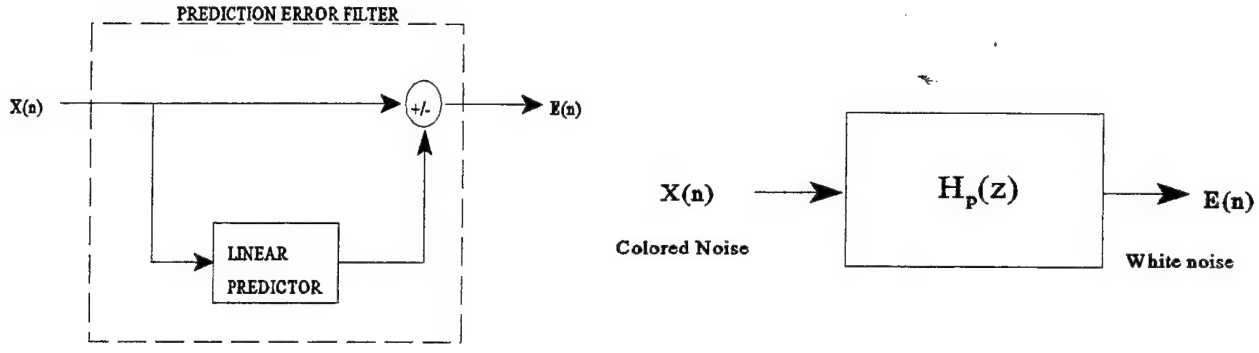

Figure 7.1: Prediction Error Filter.

The transfer function for a PEF can be represented by a finite impulse response filter (FIR) of the form:

$$H_p(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}. \qquad (7.3)$$

If the order p of the PEF is chosen sufficiently large, the output error terms $\epsilon(n)$ will be orthogonal and of constant variance, and therefore $\epsilon(n)$ will be white noise [32].

To construct the pre-whitening filter, the transfer function of the PEF is selected to be the inverse of the AR process that models the input colored noise. The output of the PEF will be a colored version of the signal in additive *white noise.* Figure 7.2. displays the process.

Selecting the model order to use for the AR model and thus also the pre-whitening filter is a difficult problem, since theorectical criteria are known to provide inaccurate results when applied to data that is not truley generated from and AR process.

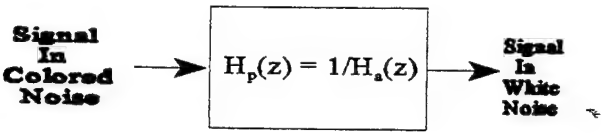

Figure 7.2: Pre-whitening Filter.

A number of different estimators are available to calculate the model order, and they are based on locating the minimum in a quantity related to the prediction error variance. One such estimator is Akaike's information-theortic criteria (*AIC*), and is described by the equation:

$$AIC(P) = N \cdot ln(\sigma_p^2) + 2P\,, \qquad (7.4)$$

where N is the length of the data, P is the model order and σ_p^2 is the prediction error variance obtained at that order. The *AIC* provides a distinct minimum at the optimum model order P. This is the criteria chosen for deciding the AR model order in this study. [32]

2. Normalizing the Noise

Recall that the threshold values are computed as a multiple of the estimated noise standard deviation σ. Once σ has been estimated, the input data can be scaled such that the noise appears at unit variance. This permits the threshold value to be computed independent of the data. Scaling of the input to produce N(0,1) noise is accomplished by using the *normnoise.m* function of the Wavelab.700 toolbox [10] and is required pre-processing for use of all denoising tools.

3. Segmenting the Data

Segmenting the data into blocks prior to processing serves two purposes. First, it provides a method to handle extremely long data records, and allows for the technique to be extended to the case of a continuous data stream. Second, it provides a means to adjust the parameters of the algorithm to account for changes in the data stream over time. For example, the noise estimates and the selection of the best basis can be "updated" for each segment.

Each data record analyzed in this study was segmented into dyadic length blocks of 2048 (2^{11}) points. Dyadic length blocks were chosen simply to avoid the need to zero pad the data prior to wavelet-based dyadic decomposition. Longer or shorter lengths could be used with equal results, however lengths of less than a few hundred points were found to perform poorly. This is attributed to the statistical nature of the threshold calculations and their assumption of large sample sizes. In addition, at a typical acoustic data sampling rate of 8 kHz, 2048 points represents approximately 0.25 seconds of acoustic data. This is a sufficiently short interval to assume the ocean noise to be stationary.

4. Signal Decomposition

The choice of a wavelet basis plays an important role in the results obtained by the analysis. Choosing the proper wavelet (i.e., the basis best matched to the signal characteristics) can have drastic effects on the overall performance of the denoising technique. Unfortunately, there is no single best wavelet basis for all signals nor is there a

perfect selection criterion other than direct comparison of results. Current research is still investigating how to best choose a mother wavelet from various libraries of basis functions [20,33].

The underwater transient data to be studied here can be separated into two broad classes. Class I data which contains man-made short duration broadband pulse transients, and class II data which contains primarily harmonic signals produced by biologics. Based on the notion that class I signals can be efficiently decomposed by the WPT (using the Symmlet 8 wavelet) and that class II signals can be efficiently decomposed by the CPT, each block of segmented data is decomposed by *both* the CPT and WPT. The best basis of each of these sets of transformed data is determined via the Best Basis algorithm. The Shannon entropy of the two resulting bases is compared, and the basis with the lowest entropy is selected as the decomposition for that segment.

5. Thresholding

The algorithm permits selection of either of the three threshold methods (soft, hard, or semi-soft). It also permits use of any of the threshold value calculations (T_u, T_S, T_M, T_H). However, in the cases studied here, the combinations of soft thresholding at value T_M, or hard thresholding with the threshold value T_u proved to be the most effective.

6. Reconstruction

Each cleaned segment is individually transformed back to the signal domain and the segments are weighted and overlapped to allow for smooth reconstruction. The 2048 point segments are overlapped 256 points or 12.5% of the total block length. Figure 7.3

displays the segmentation and reconstruction process. Figure 7.4 is a block diagram of the denoising algorithm.

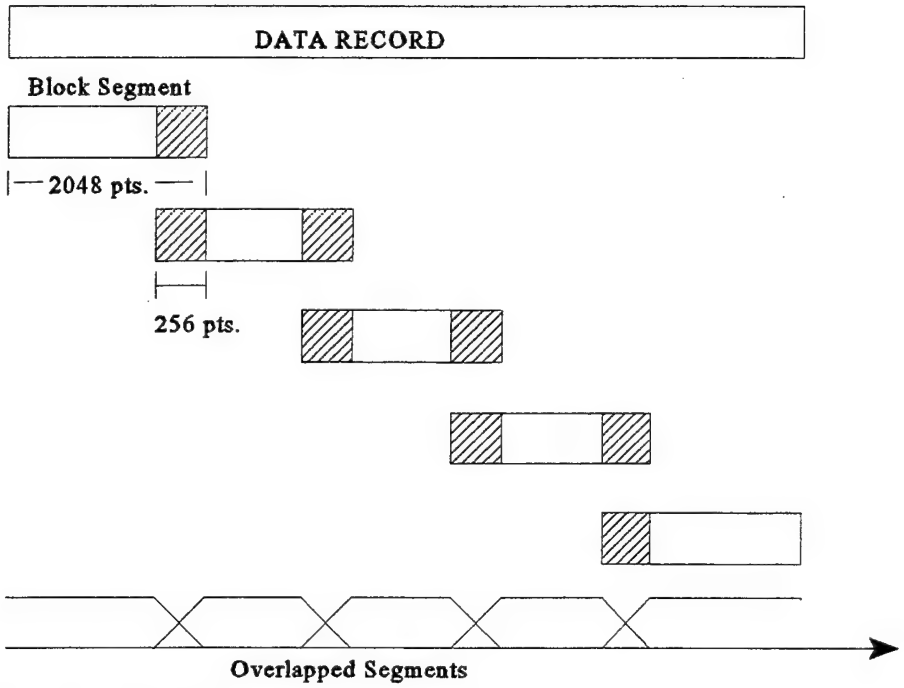

Figure 7.3: Data segments and reconstruction.

B. RESULTS

Figures 7.5 through 7.10 display the results of the denoising procedure applied to three acoustic transients. The parameters of the algorithm were the same in each case, using block segment lengths of 2048 points, and a decomposition depth of eight levels (i.e., $J = 0,1,...,8$). All data records were normalized to have unit maximum amplitude and the time axis represents the data sample number. Each of the spectrogram frequency axis display the frequency normalized to the sampling rate. No pre-whitening transform was

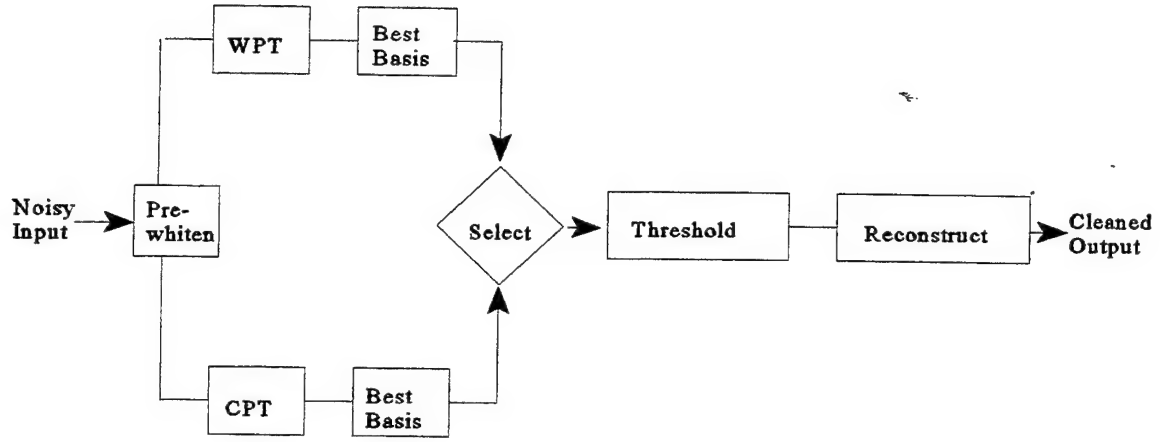

Figure 7.4: Denoising algorithm block diagram.

applied to the input data since representative noise-only samples were not available for these signals.

Figures 7.5 and 7.6 display the results of denoising a sperm whale echo using soft thresholding and the minimax threshold value T_M. Figure 7.5 shows the time series, and Figure 7.6 shows the spectrograms, of both the noisy and the cleaned sperm whale echo. The whale echo is composed primarily of *spiky impulsive like claps* in a modest amount of ocean noise and it is shown with additional white noise added. The two representations of the denoised output show the removal of a large amount of the input noise and good quality reproduction of the whale echo. The aural quality of the cleaned whale echo is also much improved. The background noise was virtually eliminated and each echo *clap* can be heard clearly and crisply.

Figure 7.7 and 7.8 display the results of denoising a gray whale recording using soft thresholding at the minimax threshold value T_M. The gray whale call is primarily a

series of modulated harmonic tones, in a relatively loud ocean noise background. The recording captures a series five calls. Figure 7.7 shows the time series, and Figure 7.8 shows the spectrograms, of the both the noisy and the cleaned gray whale calls. The figures show that a significant reduction in the noise was achieved while retaining the the prominent harmonic features of the original whale calls. The aural quality of the cleaned signal is virtually without background noise, however some distortion is audible. The audio distortion takes the form of a short, but noticible *smearing,* during the beginning of each of the one of the three separate calls.

Figure 7.9 and 7.10 display the results of denoising a humpback whale song using soft thresholding at the minimax threshold value T_M, and applying eight cycle spins of the input data. The bottom plot of Figure 7.9 displays the difference between the noisy and clean versions of the signal, and shows that primarily noise was extracted from the original input. The humpback whale song consists of four separate short tones, in a modest amount of background noise. Both the time series and the spectrograms display the prominent features of the song which are preserved by the denoising step. The aural quality of the cleaned song is also good. The only distortion audible in the cleaned version of the song is in the remaining background noise, which was not uniformly eliminated, and has a slightly *digitized* sound. A small number of cycle spins of the data was found to reduce this audible effect, and it is possible that more "spins" would have enhanced the sound quality further.

On each of the above biologic transients extremely positive results were achieved even though the pre-whitening filter was not used. This success is attributed to the relatively high SNR, and the "nearly white" nature of the noise. Spectrogram information indicates that the noise contained in the humpback and sperm whale data is wideband, while the noise in the gray whale data is more lowpass in nature. Applying the denoising algorithm to the gray whale data leads to underestimation of the noise level in the signal region.

The hard thresholding method using the universal threshold value T_u was also attempted on the above data sets, and produced similarly good results. However, hard thresholding at the higher T_u, typically displayed slightly more appealling visual plots (i.e., cleaner appearing time series) but slightly worse audio results (i.e., slightly more sound distortion) on these signals.

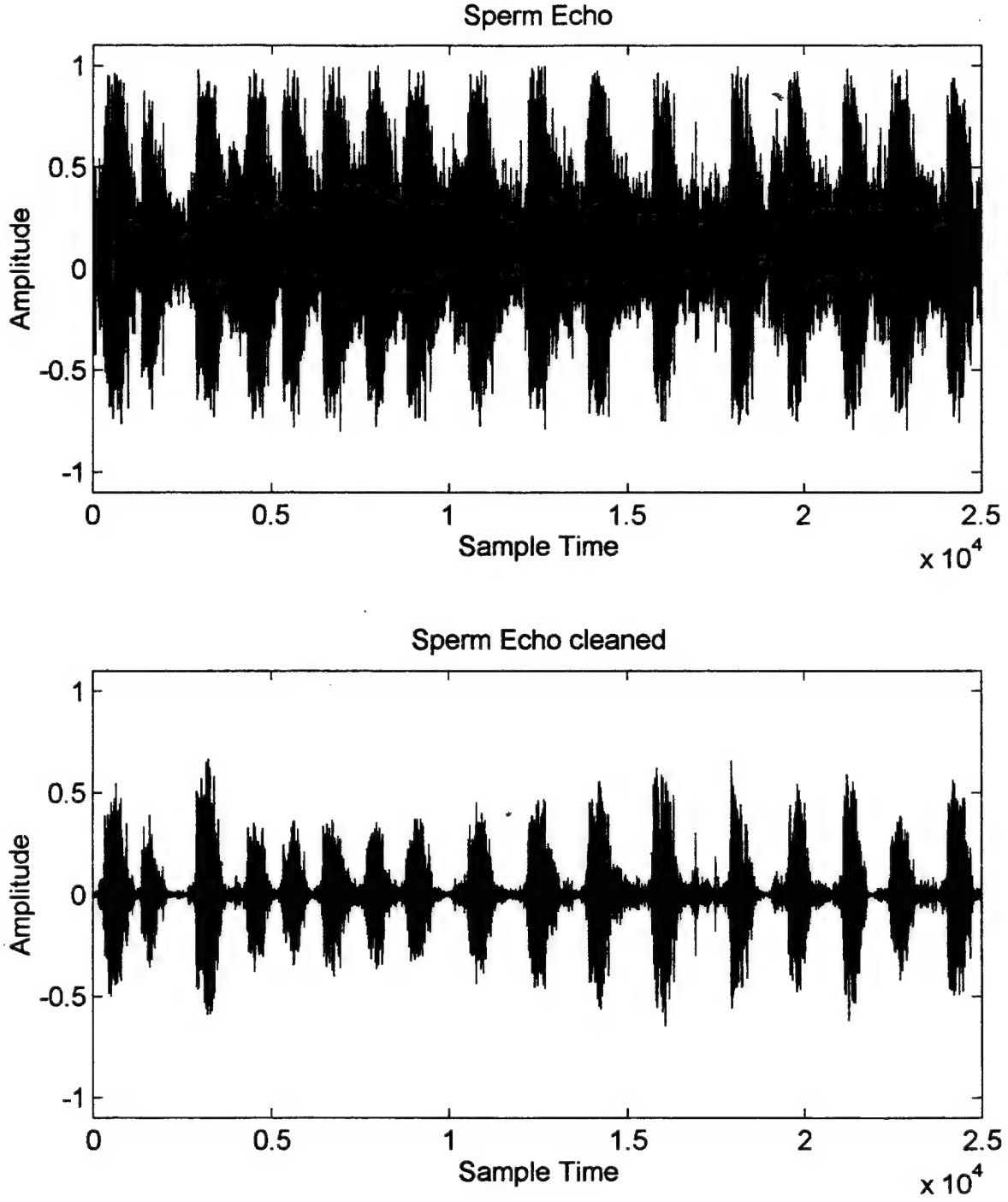

Figure 7.5: Time series of the results of denoising a sperm whale echo.

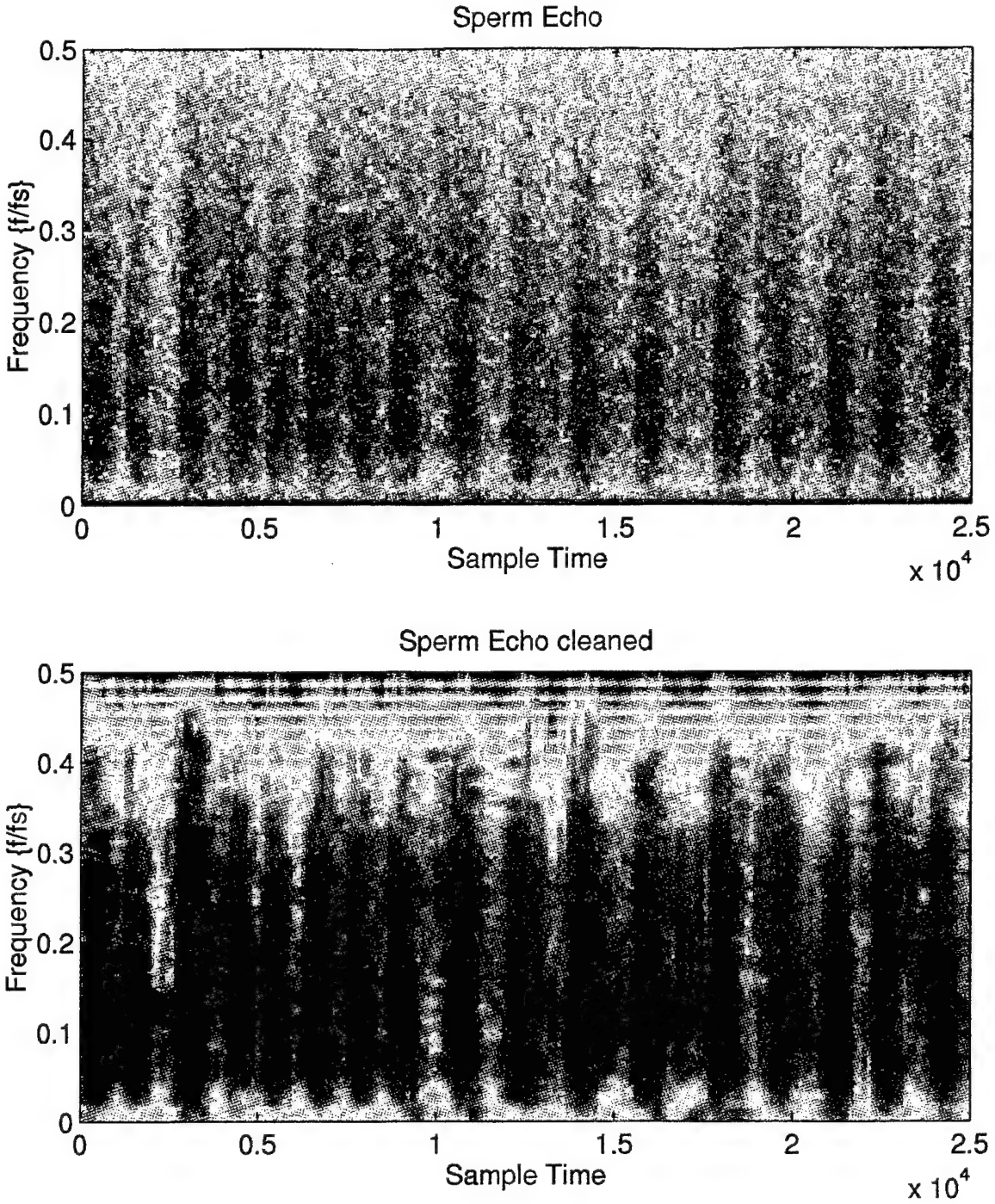

Figure 7.6: Spectrograms of the results of denoising a sperm whale echo.

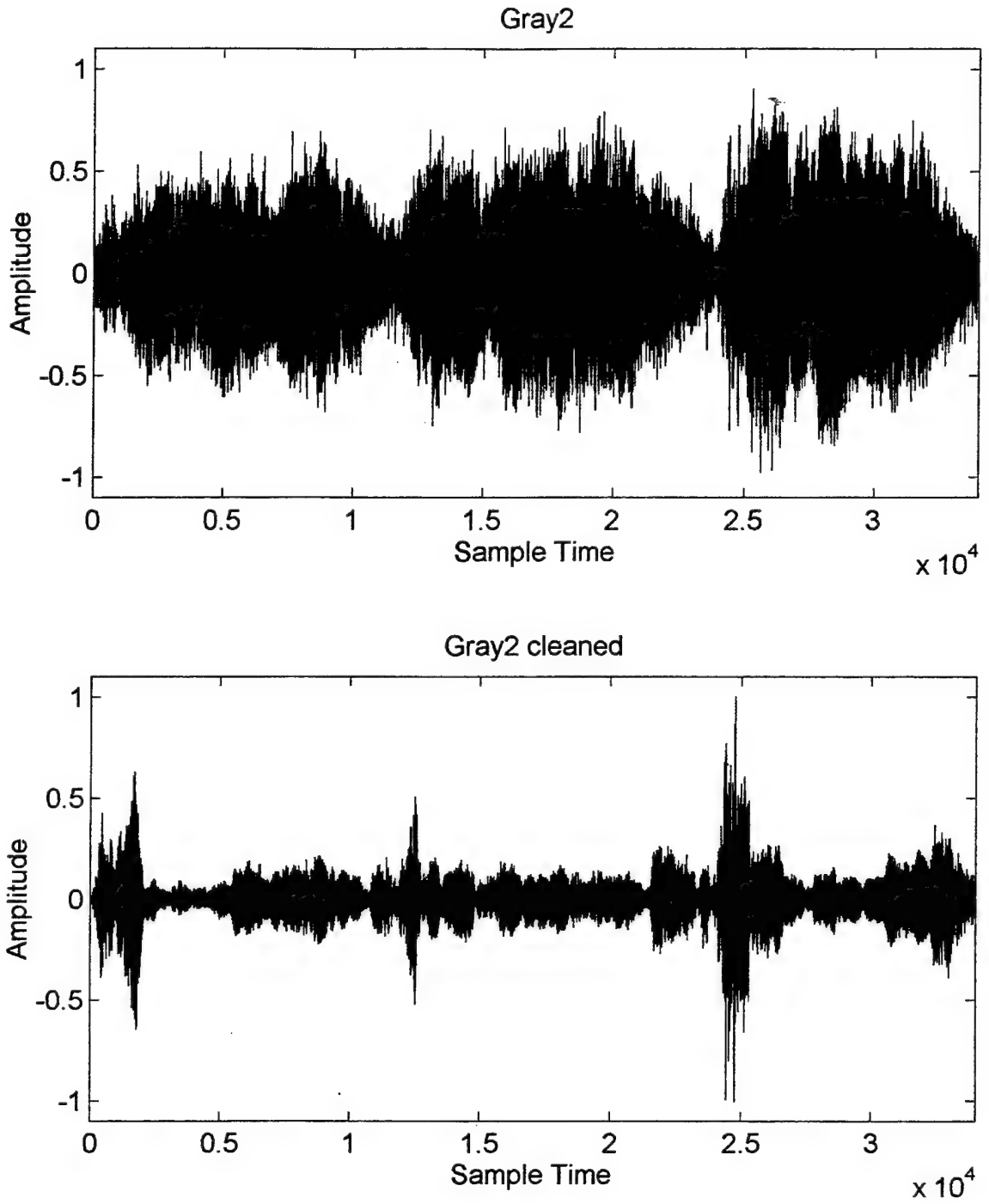

Figure 7.7: Times series results of denoising a gray whale call.

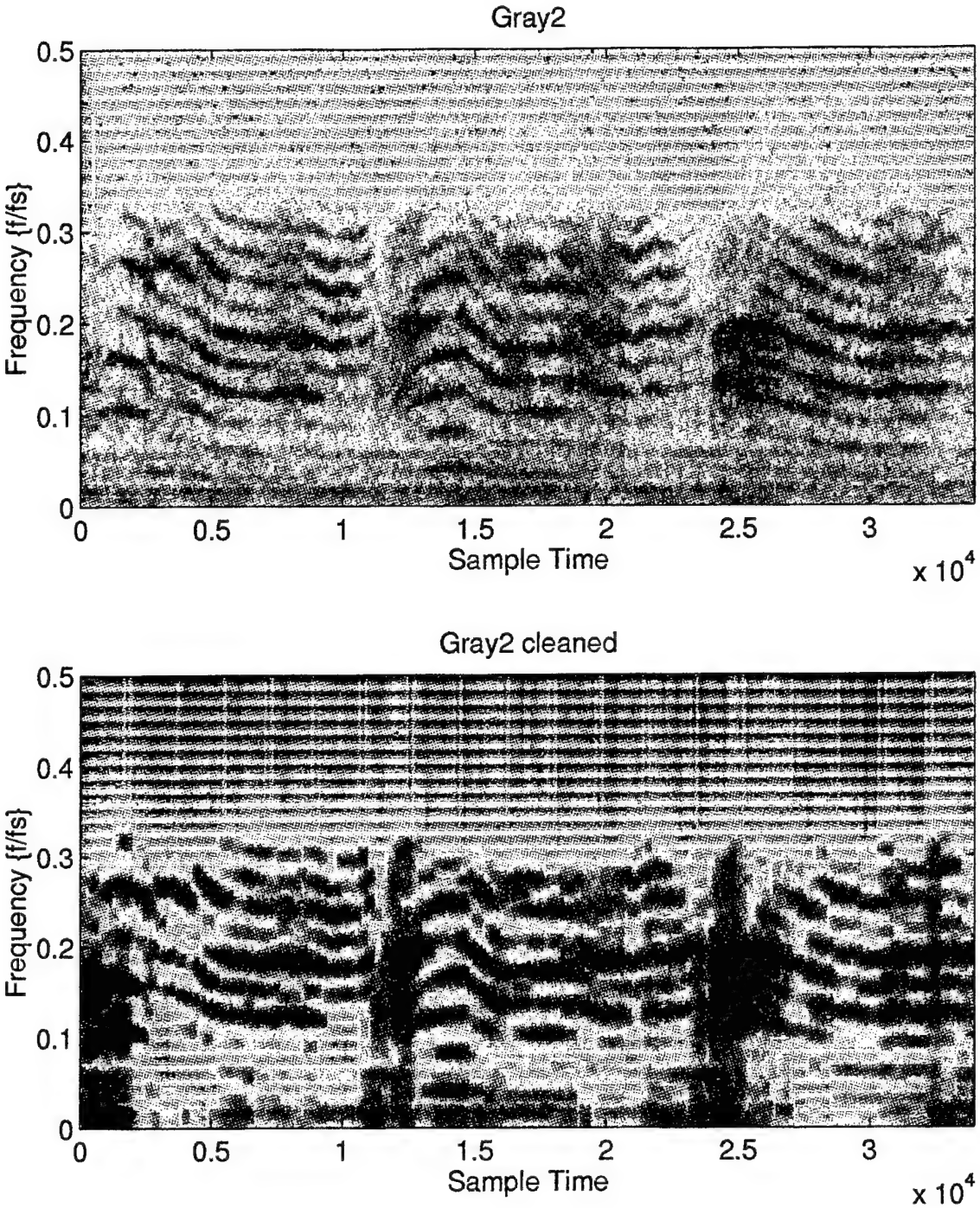

Figure 7.8: Spectrograms of the results of denoising a gray whale call.

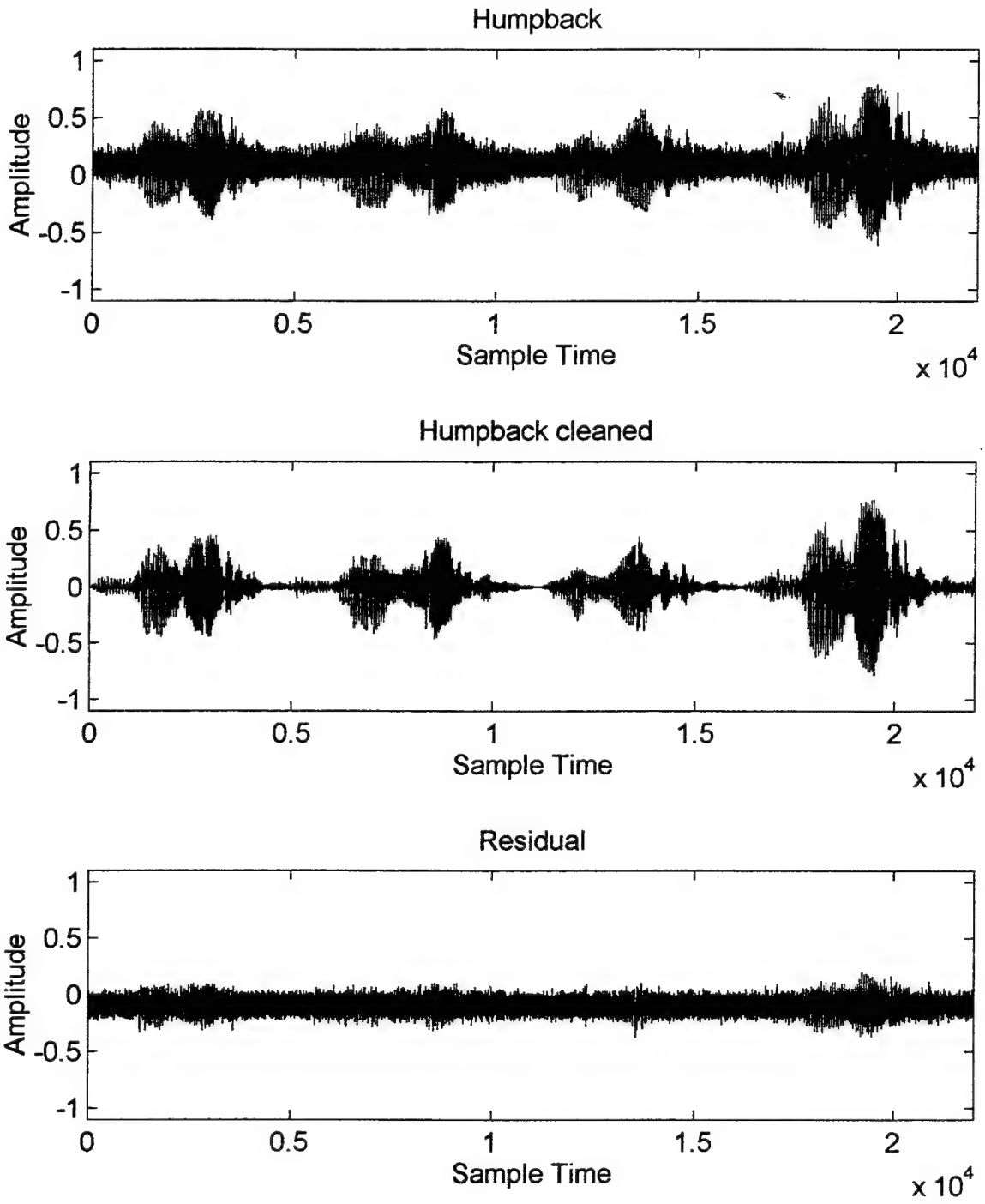

Figure 7.9: Time series results of denoising a humpback whale song.

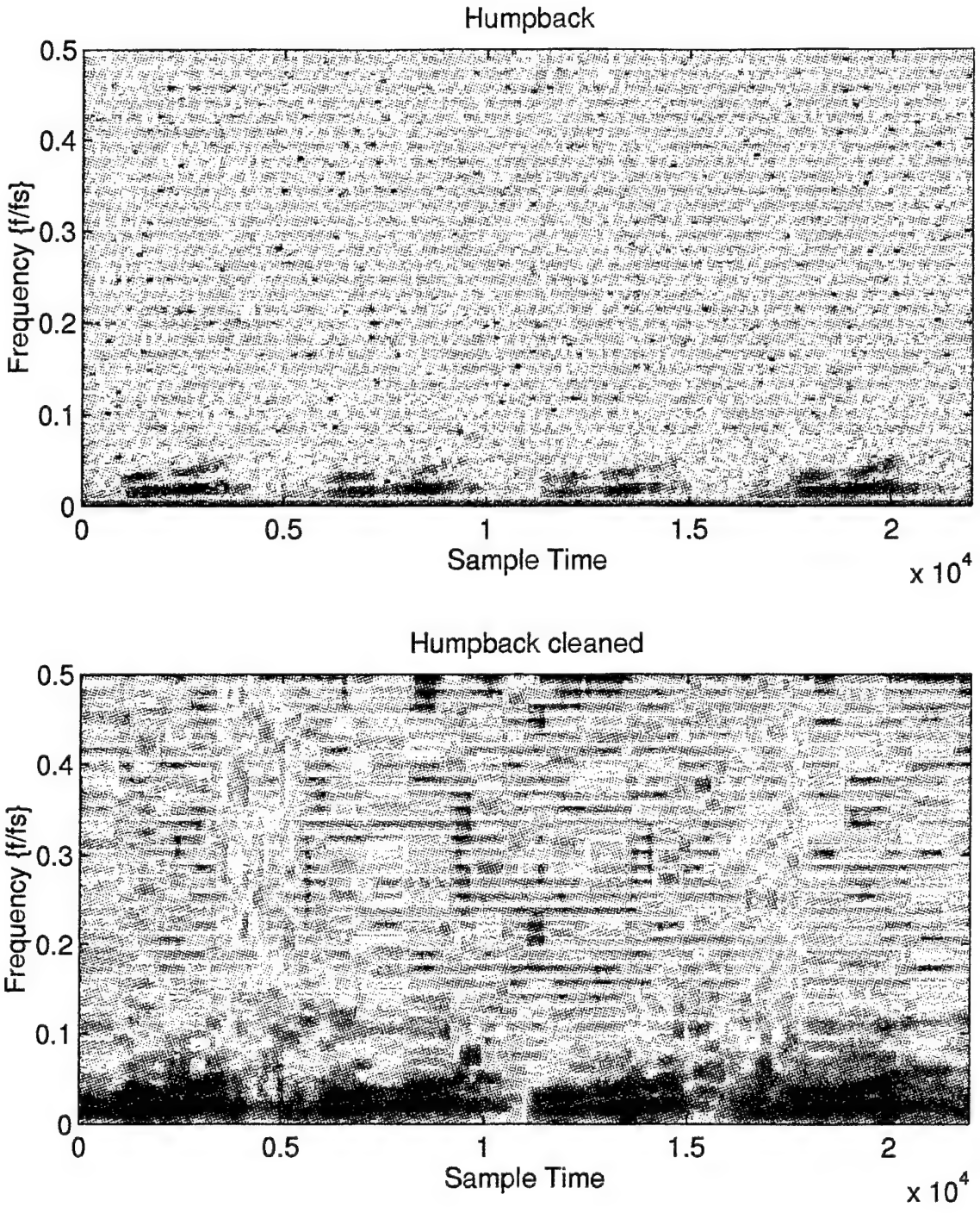

Figure 7.10: Spectrograms of the results of denoising a humpback whale song.

C. COMPARISON WITH SHORT-TIME WIENER FILTER

1. Wiener Filtering

The optimum linear filter for estimating a deterministic sequence $s(n)$ from a noisy input data set $x(n)$ is a Wiener filter. It is based on the statistical characteristics of the input and produces a minimum mean square error output. Although the Wiener filter is usually used for stationary signals, Frack [31] developed a *short-time* Wiener filter procedure which filters the data after segmenting it into short-time (essentially) stationary blocks. The problem solved by the Wiener filter is the estimation problem given by:

$$x(n) = s(n) + n(n), \qquad (7.5)$$

where $n(n)$ represents the noise portion of the input. The Wiener solution is found by estimating the input data correlation function $R_x(l)$ and the noise correlation function $R_n(l)$, and solving the Wiener-Hopf linear equations given by:

$$\sum_n R_x(n-l)\, h(n) = R_{sx}(l) = R_x(l) - R_n(l), \qquad (7.6)$$

for the unknown filter weigths $h(n)$. In the approach taken by Frack [31], the filter weights were computed for each segment of the input data. So, $R_x(l)$ is computed on each segment, $R_n(l)$ is computed from noise only data, and $R_{sx}(l)$ is found using the relation $R_{sx}(l) = R_s(l) = R_x(l) - R_n(l)$, since signal and noise are assumed uncorrelated. The result is a set of optimum filter weights for each segment of the input.

2. Comparison

Figure 7.11 displays the signal used for comparison of the short-time Wiener filter and the wavelet-based thresholding. It is a short duration synthetic acoustic transient that is emersed in successively greater amounts of Gaussian colored noise. The signal consists of three short duration transient "clicks", one large click followed by two smaller ones. The entire signal duration is approximately four seconds, and has been digitized into 16384 (2^{14}) samples. The signal was constructed of three separate decaying exponentials S_1, S_2, and S_3, given by:

$$S_1 = \sin(2\pi k_1 /4) \exp(-k_1/200) \qquad k_1 = 1,2,...,256\,, \qquad (7.7)$$

$$S_2 = \tfrac{1}{2} \sin(2\pi k_2 /4) \exp(-k_2/200) \qquad k_2 = 1,2,...,200\,, \qquad (7.8)$$

$$S_3 = \tfrac{1}{3} \sin(2\pi k_3 /4) \exp(-k_3/200) \qquad k_3 = 1,2,...,128\,, \qquad (7.9)$$

where S_1 and S_2 were separated by 2000 points, S_2 and S_3 were separated by 1000 points respectfully. The data was processed identically for both the Wiener filter and the wavelet-based denoising methods. The pre-whitening transform was applied using an AR model of order 10, as determined by the *AIC* criteria (defined by Equation 7.4), and was based on a noise-only sample of 4096 data points.The data was segmented in to blocks of 2048 points. {Note that the Wiener filter also requires relatively large block sizes on the order of 1000 or more points to provide good estimates of the correlation matrices.}

A Wiener filter length of 40 was chosen based on the criteria presented in reference [31]. The Wavelet denoising was performed using hard thresholding at the universal threshold value, and employing eight levels of dyadic decomposition.

Adjacent to each signal plot in Figure 7.11 are the denoising results obtained by the Wiener filter and the wavelet-based thresholding. In each of these relatively high SNR cases both methods recover the signal without difficulty, and both methods reproduce all three clicks. The *MSE* (shown above each plot) of the two methods was comparable, with the Wiener filter performing slightly better. However, the wavelet thresholding method produced a much cleaner appearing and less noisy sounding output, completely removing the noise between signal clicks.

Figure 7.12 shows the relative performance of the two methods at lower SNRs. Note here that both methods fail to detect the smaller and later two clicks. However, the wavelet method out-performs the Wiener filter both in terms of *MSE* and visual appearance of the time series, and continues to extract the largest of the three clicks down to -15 dB SNR, whereas the Wiener filter loses the signal entirely.

Soft thresholding at the lower T_M value was also attempted, and resulted in poorer, low SNR performance by the wavelet method on this data. A variety of Wiener filter lengths were also tried (orders as high as 90, and as low as 30) with similar results in each case. Additionally, other parameters were varied, including the noise AR model order, and data segment lengths, and were found to produced similar results.

Finally, similar findings were also obtained using other sets of ocean acoustic data, and further results are presented in reference [34].

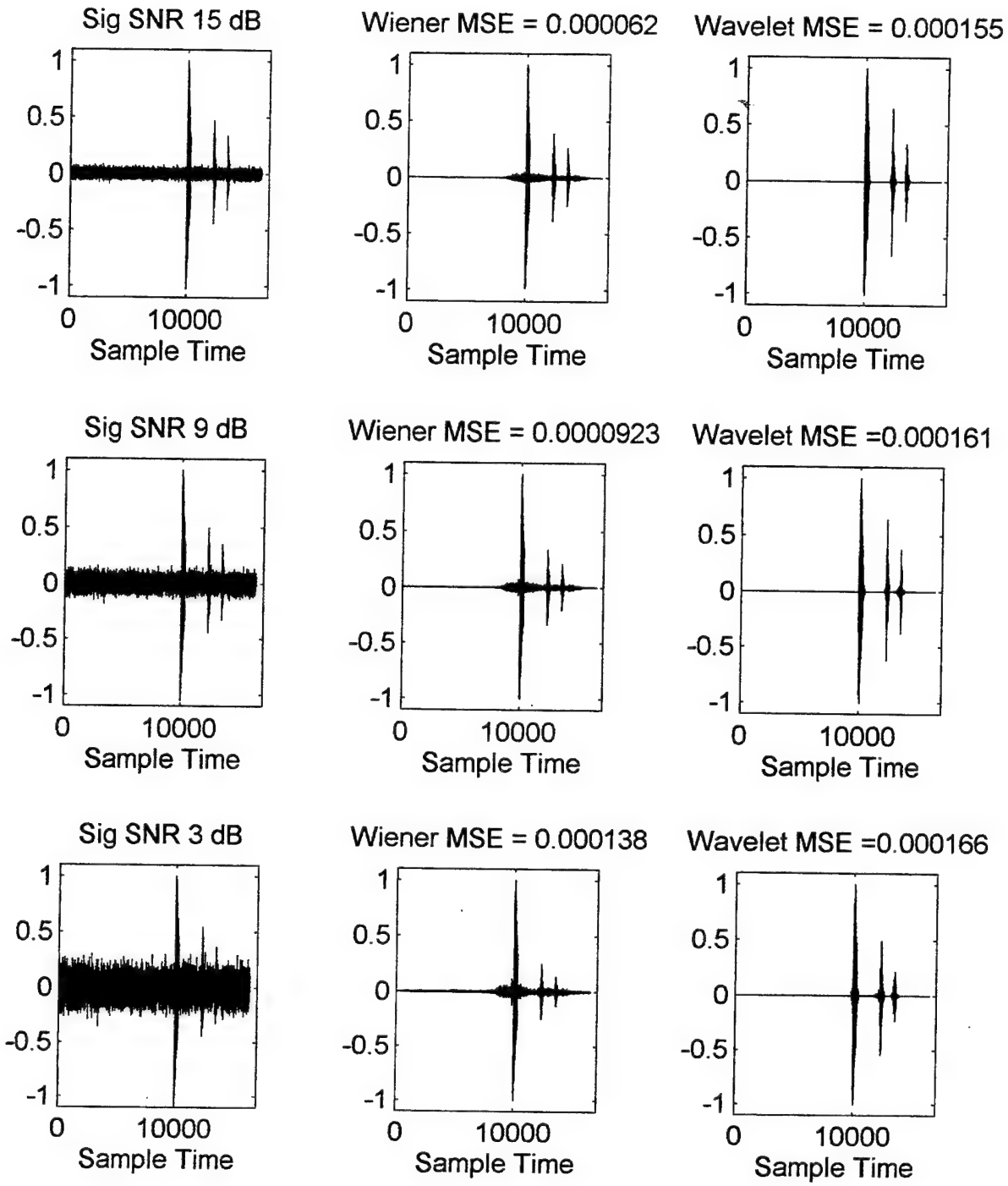

Figure 7.11: Comparison of Wiener filter and wavelet-based thresholding at high SNRs.

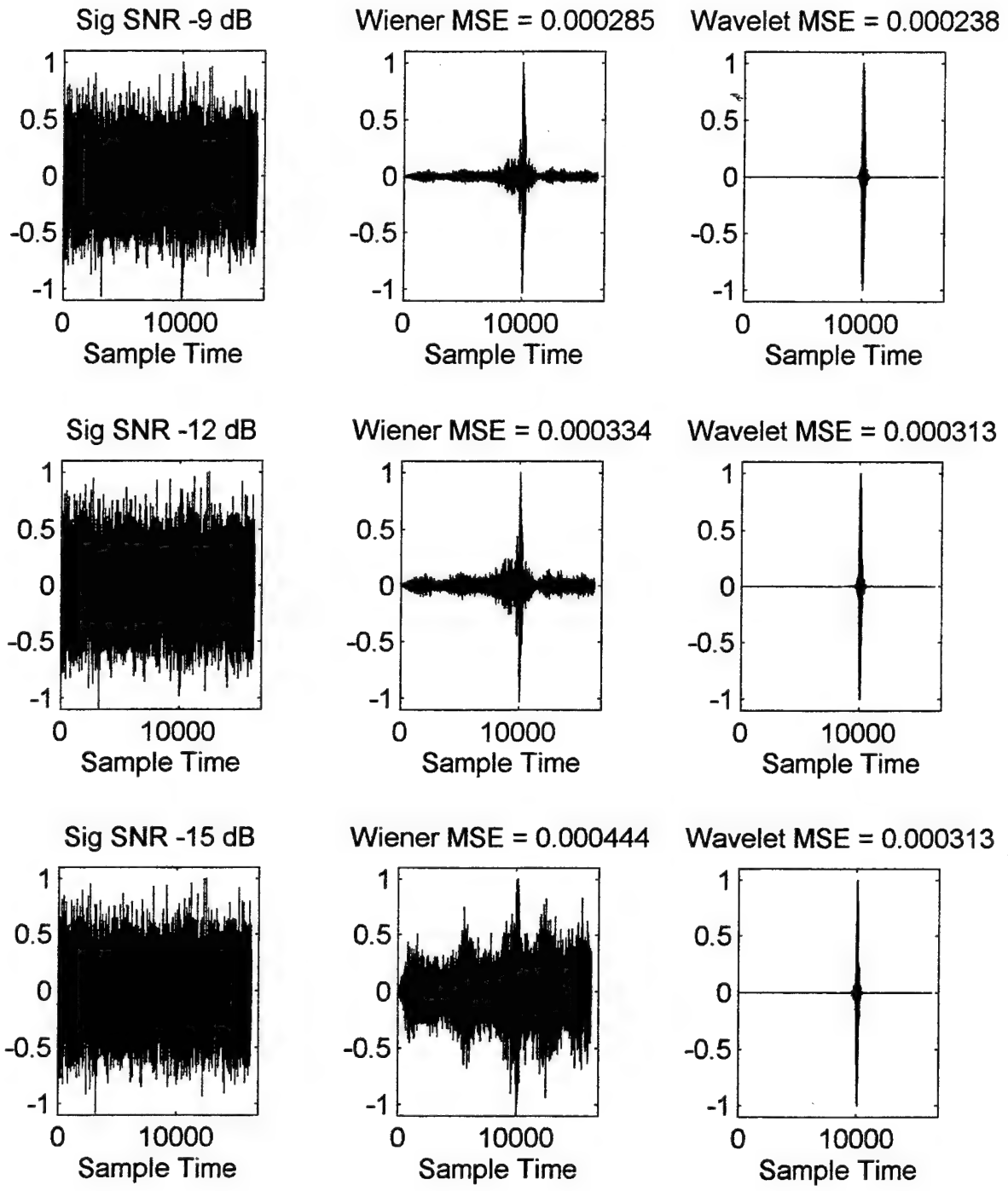

Figure 7.12: Comparison of Wiener filter and wavelet-based thresholding at low SNRs.

VIII. CONCLUSIONS

In this thesis we have compared a number of wavelet-based denoising techniques and applied them to the problem of noise removal from ocean acoustic transients. The denoising procedure outlined made use of both the wavelet packet and cosine packet transforms, the thresholding techniques of Donoho et.al. [2], and the Best Basis algorithm of Wickerhauser and Coifman [24]. This method was shown to perform well on both broadband transient and narrowband harmonic signals even in low SNR colored noise environments. The implemented scheme provides a viable denoising procedure for a wide variety of acoustic transients.

In particular, the contrast of wavelet-based denoising schemes demonstrated the superiority of wavelet packet decompositions to that of the wavelet transform, the robust nature of the Symmlet 8 wavelet basis, and the general applicability of the universal threshold value. Additionally, cycle spinning of the input data was shown to improve the denoising performance as well as provide a means to combat the translation invariance of the wavelet-based transformations. The review of basis selection demonstrated that the use of the Shannon entropy criterion provides a accurate measure by which to compare the efficiency of two bases, and that the basis search (typically the most time consuming portion of the algorithm) need not be exhaustive and can be truncated to some reasonable number of levels (e.g., eight levels for a data set of 2^{11} samples). Finally, the performance of the denoising algorithm to a number of diverse signals was presented, and a comparison with a short-time Wiener filter was conducted.

The wavelet-based denoising technique compared favorably to the short-time Wiener filter. At high SNRs the two methods produced similar results, with the Wiener filter providing a slightly better *MSE,* and the wavelet denoising providing a cleaner visual time series and less noisy aural output. At lower SNRs, the wavelet denoising out-performed the Wiener filter in terms of *MSE,* visual appearance, and in its ability to extract the largest signal peak at a lower SNR.

The ideas and results presented in this thesis provide a number of opportunities for future research and investigation. In the remaining paragraphs a few topics that could expand this work are suggested.

Wavelet-based threshold denoising can be viewed as an adaptive compression scheme that selects the number of reconstruction coefficients to be retained based on an estimate of the input noise level. In this view of the process, threshold denoising is a natural precursor to signal classification since in principal, it reduces the complexity of the signal by eliminating only non-interesting features (i.e., the noise). The design of such classifier, that operates in the wavelet domain is a possible area of future study. An interesting consequence of such a classifer is that signal features could be extracted and identified in the wavelet transform domain, precluding the need for aural reconstruction of the signal.

A second possible area of study would be to incorporate additional basis sets into the procedure to complement the local cosine and Symmlet 8 bases. For example, an interesting choice would be a wavelet basis composed of decaying sinusoids. Futhermore,

an entire "library" of basis sets could be applied to both the denoising and classification of acoustic transients, similar to the methods described by Saito [20].

The cycle spinning technique is a computational burden since it requires multiple applications of the algorithm to be conducted and then the results averaged. Another approach might be to calculate all possible circular shifts of the input data in the wavelet packet best basis, using a version of the fast algorithm of Beylkin [17]. This would essentially be the same process used to perform the undecimated discrete wavelet transform with the exception that the decomposition would occur according to the selected wavelet packet best basis. This scheme would produce a fast translation invariant, wavelet packet decomposition, and should improve the denoising performance.

LIST OF REFERENCES

[1] O. Rioul and M. Vetterli, " Wavelets and Signal Processing," *IEEE Signal Processing Magazine*, October, 1991

[2] D. Donoho and I. Johnstone, "Adapting to Unknown Smoothness Via Wavelet Shrinkage," *Journal of the American Statistical Association*, December 1995, Vol. 90, No. 432, Theory and Methods

[3] R. Urick, *Ambient Noise in the Sea*, Peninsula Publishing, Los Altos, California, 1986

[4] L. Kinsler, A. Coppens, A. Frey, and J. Sanders, *Fundamentals of Acoustics*, John Wiley & Sons, Inc., New York, 1982

[5] The Mathworks, Inc., *Matlab Reference Guide*, Massachusetts, 1992

[6] J. Devore, *Probability and Statistics for Engineering and the Sciences*, Wadsworth, Inc., California, 1991

[7] S. Haykin, *Communication Systems*, John Wiley & Sons, Inc., New York, 1994

[8] N. Morrison, *Introduction to Fourier Analysis*, John Wiley & Sons, Inc., New York, 1994

[9] J. Sadowsky, "The Continuous Wavelet Transform: A Tool for Signal Investigation and Understanding," Johns Hopkins APL, Technical Digest, Vol. 15, No. 4, 1994

[10] J. Buckheit, S. Chen, D. Donoho, and J. Scargle, "Wavelab.700," http://www.wavelab/ playfair.stanford.edu, 1996

[11] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pat. Anal. Mach. Intell*, Vol. 11, 1989, pp. 674-693

[12] J. Proakis and D. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*, Macmillan Publishing Company, New York, 1992

[13] V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A. K. Peters, Ltd., Massachusetts, 1994

[14] R. Loe, K. Anderson, and K. Jung, "Comparitive Analysis Results for Underwater Transient Classification," *SPIE*, Vol. 2242, pp. 815-823,Wavelet Applications, 1994

[15] S. Del Marco, J. Weiss, and K. Jagler, "Wavepacket-Based Transient Signal detector Using a Translation Invariant Wavelet Transform," *SPIE*, Vol. 2242, pp. 792-802, Wavelet Applications, 1994

[16] R. Coifman and D. Donoho, "Translation-Invariant Denoising," Internal Report, Department of Statistics, Stanford University, 1995

[17] G. Beylkin, "On the Representation of Operators in Bases of Compactly Supported Wavelets," *SIAM Journal Numerical Analysis*, Vol 29, pp. 1716-1740, 1992

[18] M. Lang, H. Guo, J. Odegard, and C. Burrus, "Nonlinear Processing of a Shift-Invariant DWT for Noise Reduction, *SPIE*, Vol. 2491, pp.640-651, 1995

[19] M. Cody, " The Wavelet Packet Transform," *Dr. Dobbs Journal*, April 1994

[20] N. Saito, *Local Feature Extraction and Its Applications Using a Library of Bases*, PhD Dissertation, Yale University, 1994

[21] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Inc., New Jersey, 1996

[22] R. Martins, *Speech Compression Using Cosine Packet Decomposition*, MSEE Thesis, Naval Postgraduate School, Monterey, California, March 1995

[23] The Mathworks Inc., *Wavelet Toolbox User Guide*,Massachusetts, 1996

[24] R. Coifman and M. Wickerhauser, "Entropy Based Algorithms for Best Basis Selection," *IEEE Trans. On Information Theory*, Vol. 3b, No. 2, March 1992

[25] D. Donoho and I. Johnstone, " Ideal Spatial Adaptation by Wavelet Shrinkage," *Biometrika*, Vol. 81, No. 3, pp 425-455, 1994

[26] S. Kenny, *Mathematics of Statistics*, D. Van Nostrand Company, New York, 1942

[27] M. Leadbetter, G. Lindgren, and H. Rootzen, *Extremes and Related Properties of Random Sequences and Processes*, Springer-Verlag, New York, 1983

[28] C. Stein, " Estimation of the Mean of a Multivariate Normal Distribution," *The Annals of Statistics*, Vol. 9, pp. 1135-1151, 1981

[29] A. Bruce and H. Gao, "Waveshrink: Shrinkage Functions and Thresholds," Technical Report, StatSci Division of MathSoft, Inc., 1995

[30] I. Johnstone and B. Silverman, "Wavelet Threshold Estimators for Data with Correlated Noise," Department of Statistics, Stanford University, December 21, 1994

[31] K. Frack, *Improving Transient Signal Synthesis Through noise Modeling and Noise Removal*, MSEE Thesis, Naval Postgraduate School, Monterey, California, March 1994

[32] C. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, Inc., New Jersey, 1992

[33] S. Chen and D. Donoho, " Basis Pursuit," Technical Report, Department of Statistics, Stanford University, 1994

[34] R. J. Barsanti, M. P. Fargues, R. Hippenstiel, "Investigations in the Denoising of Ocean Acoustic Data using Wavelet-Bas101ed Techniques", NPS Technical Report, Jan. 1994

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2. Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, CA 93943-5101	2
3. Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
4. Prof. Monique P. Fargues, Code EC/Fa Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	5
5. Prof. Ralph Hippenstiel, Code EC/Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
6. Prof. Robert Keolian, Code PH/Kn Engineering Acoustics Academic Group Naval Postgraduate School Monterey, CA 93943-5117	1
7. Prof. James Eagle, Code OR/Er Undersea Warfare Academics Group Naval Postgraduate School Monterey, CA 93943-5219	1

8. Mr. Michael Shields, Code EC/Sl 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

9. LCDR Robert J. Barsanti Jr., Code EC/Br 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121